

# Aufkleber für Einbanddeckel

STUDIENSEMINAR  
FÜR DAS LEHRAMT FÜR  
DIE SEKUNDARSTUFE II  
DETMOLD

Verfahren zur Mustersuche unter  
besonderer Berücksichtigung der  
Verbesserung von Algorithmen

KOPIE

Schriftliche Hausarbeit gem. § 17 OVP  
vorgelegt von: StRef Hubert Hinrichs  
Lehramt: Sekundarstufe II / I (24)  
Fach: Informatik  
Einstellungstermin: 01.02.1997  
Bearbeitungszeitraum: 06.03.98-06.06.98  
zuständiger Fachleiter: StD Lothar F. Carl

STUDIENSEMINAR  
FÜR DAS LEHRAMT FÜR DIE SEKUNDARSTUFE II  
DETMOLD

Verfahren zur Mustersuche unter besonderer  
Berücksichtigung der Verbesserung von Algo-  
rithmen

KOPIE

Schriftliche Hausarbeit gem. § 17 OVP  
vorgelegt von: StRef Hubert Hinrichs  
Lehramt: Sekundarstufe II / I (24)  
Fach: Informatik  
Einstellungstermin: 01.02.1997  
Bearbeitungszeitraum: 06.03.98-06.06.98  
zuständiger Fachleiter: StD Lothar F. Carl

## Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>2</b>
1.1 Wahl des Themengebiets .....	2
1.2 Verständnis des Themas .....	4
1.3 Planung und Darstellung der Stunden.....	5
<b>2 Planung der Unterrichtsreihe</b>	<b>6</b>
2.1 Fachwissenschaftlicher Rahmen.....	6
2.2 Intentionen der Unterrichtsreihe .....	11
2.3 Lernvoraussetzungen.....	12
2.4 Auswahl der Verfahren.....	14
2.5 Einsatz von Medien .....	15
2.6 Begründungszusammenhang .....	18
<b>3 Die erste Stunde</b>	<b>20</b>
3.1 Planung.....	20
3.2 Durchführung .....	23
3.3 Reflexion.....	25
<b>4 Die zweite Stunde</b>	<b>28</b>
4.1 Planung.....	28
4.2 Durchführung .....	32
4.3 Reflexion.....	34
<b>5 Die dritte Stunde</b>	<b>35</b>
5.1 Planung.....	35
5.2 Durchführung .....	37
5.3 Reflexion.....	39
<b>6 Die vierte Stunde</b>	<b>41</b>
6.1 Planung.....	41
6.2 Durchführung .....	44
6.3 Reflexion.....	47
<b>7 Die fünfte Stunde</b>	<b>48</b>
7.1 Planung.....	48
7.2 Durchführung .....	51
7.3 Reflexion.....	53
<b>8 Die sechste Stunde</b>	<b>54</b>
<b>9 Die siebte Stunde</b>	<b>55</b>
9.1 Planung.....	55
9.2 Durchführung .....	58
9.3 Reflexion.....	61
<b>10 Gesamtreflexion</b>	<b>62</b>
<b>11 Anhang A: Materialien zur 1. - 4. Stunde</b>	<b>66</b>
<b>12 Anhang B: Materialien zur 5. - 7. Stunde</b>	<b>74</b>
<b>13 Anhang C: Listing des erstellten Programms</b>	<b>82</b>
<b>14 Literaturverzeichnis</b>	<b>85</b>

# 1 Einleitung

## 1.1 Wahl des Themengebiets

*„A fundamental technique used in computer science is to search for a specific substring in a larger body of text. Algorithms that do this rank with sorting algorithms as cornerstones of software methodology.“*

(Daniel M. Sunday: A Very Fast Substring Search Algorithm)

*„Dank immer schnellerer Rechner bemühen sich immer weniger Programmierer um effiziente Programme - schade eigentlich, denn dadurch bleibt unterm Strich alles beim alten. Dabei ist selbst die einfache Aufgabe, ein Wort in einem Text zu suchen, mit ein wenig Überlegung und ein paar zusätzlichen Programmzeilen um das Sechsfache zu beschleunigen. Das lohnt sich doch, oder?“*

(Michael Tamm: Blitzfindig)

Die Mustersuche findet in vielen Gebieten der Informationsverarbeitung Anwendung: Textverarbeitungssysteme und Editoren, Literaturdatenbanken, Internet (Wide Area Information System, Indizierungsautomaten wie CyberSitter oder NetNanny), Betriebssysteme, die die Möglichkeit bieten, nach Dateien mit bestimmtem Textinhalt zu suchen (grep in Unix, Explorer in Windows95) oder Virens Scanner, die nach einem verdächtigen Binärmuster etwa im Boot-Sektor einer Diskette suchen. Dabei kennen die meisten Computernutzer zumindest die Möglichkeit des Suchens oder des kombinierten Suchens und Ersetzens bei ihrem Textverarbeitungssystem.

Es ist daher naheliegend, nach einem zugrunde liegenden Verfahren zu fragen. Ebenso ist es danach aus informatischer Sicht sinnvoll, über Verbesserungen dieses Verfahrens nachzudenken. Gemeint sind hier nicht trickreiche Verbesserungen der Implementation, etwa durch bestimmte programmiersprachliche Elemente (exit, goto etc.), sondern Verbesserungen, die auf neuen Ideen oder Strategien für das Suchverfahren beruhen. Das Themengebiet ist voll von Verfahren, denen interessante, teils schwierige, teils sehr simple, aber durchschlagende Ideen zugrunde liegen.

Die erste im Rahmen dieser Unterrichtsreihe angestrebte Verbesserung ist mit Hilfe eines nur kleinen Impulses von den Schülern leicht nachzuerfinden, mit

angemessenem Aufwand zu algorithmisieren und zu implementieren und bietet darüber hinaus immensen Performancegewinn. Eine weiterreichende Ausnutzung der dort zugrunde liegenden Idee ist dann erheblich schwieriger und wird erst unter bestimmten Umständen (große Musterlänge) zu einem echten Laufzeitgewinn führen. Dies ist bei der realen Entwicklungstätigkeit im Bereich Mustersuche ebenfalls oftmals der Fall,<sup>1</sup> so daß das Themengebiet eine gute Chance bietet, mit Schülern diesen Prozeß einmal zu durchlaufen. Ebenso bietet dieses Themengebiet besonders gute Chancen, ein offenes Ende der Reihe zu gestalten, indem etwa weitere, innerhalb der Reihe nicht realisierte Ideen und Strategien ausblickartig angedacht und erörtert werden können.

Die prinzipielle Idee einer der geplanten Verbesserungen des naiven Suchverfahrens, nämlich einige Vergleiche einzusparen, indem man den Ausgang eben dieser Vergleiche durch geschickte zusätzliche Vorüberlegungen voraussieht, ist in der Informatik fundamental: Der binären Suche und entsprechenden Bäumen liegt die gleiche prinzipielle Idee des Einsparens von Vergleichen durch deren Vorausschau zugrunde. Die bei der hier angestrebten Verbesserung nötige Vorverarbeitung des Musters kommt in ähnlicher Form in sehr vielen Algorithmen zur Mustersuche vor, so daß die Schüler einen wesentlichen Grundzug der meisten Algorithmen in diesem Bereich kennenlernen. Der Themenbereich bietet auch viele Möglichkeiten für eigene Verbesserungsideen der Schüler, denen dann auch nachgegangen werden kann.

Außerdem bietet das Themengebiet die Möglichkeit, gefundene Ideen für Verfahren weiter "auszureizen" oder sie mit andere Ideen zu koppeln und daraus jeweils neue Verfahren zu gewinnen, was das heuristische Denken fördern kann. Die abschließende Beurteilung der entwickelten Algorithmen inklusive der Entwicklung der nötigen Testkriterien und der Testvorbereitung sind typisch informatische Tätigkeiten. Diese Beurteilung der Algorithmen läßt dann auch Rückschlüsse auf die Tragfähigkeit der jeweils dahinterliegenden Ideen im Zusammenhang mit diesem Thema zu, so daß die Schüler auch hier heuristisches Denken und die Beurteilung von Strategien üben können.

In den z.Zt. noch gültigen Richtlinien ist die dem Problem zugrunde gelegte und von den schnelleren Algorithmen wegen des freien Zugriffs benötigte Datenstruktur `Array` dem Lernbereich "Daten und Datenstrukturen" (II 2.2) zugeordnet.<sup>2</sup> Die Mustersuche in Texten wird nicht explizit aufgeführt, läßt sich aber im Lernbereich Algorithmik unter den Punkt " 'Standardalgorithmen' der DV, Suchalgorithmen" (I 5.1) einordnen.<sup>3</sup> Die Tatsache, daß das Wort Standardalgorithmen in Anführungsstriche gesetzt ist, soll wohl darauf hindeuten, daß

---

<sup>1</sup> vgl. Tamm, S.294 oben und Kasten „Theorie und Praxis“ , sowie Smith S.1072

<sup>2</sup> vgl. Richtlinien S.37

<sup>3</sup> vgl. Richtlinien S.35

dieser Bereich, auch wegen der großen Dynamik der Fachwissenschaft, inhaltlich flexibel gedacht ist.

Der in dieser Unterrichtsreihe abschließend durchzuführende Vergleich der implementierten Algorithmen ist im Lernbereich Algorithmik unter dem Stichwort "Test (...) von Algorithmen" (I 3.3) zu finden,<sup>4</sup> so daß man von einer Rechtfertigung des Themas durch die z.Zt. gültigen Richtlinien sprechen kann. Der neue Lehrplanentwurf sieht als ein Ziel des Informatikunterrichts, daß die Schülerinnen und Schüler "Lösungskonzepte in einer Programmiersprache realisieren, überprüfen und weiterentwickeln", und als Unterpunkt wird das "Optimieren von Problemlösungen" genannt.<sup>5</sup>

Dieser neue Entwurf bleibt bezüglich der konkreten Inhalte, die im Informatikunterricht zu vermitteln sind, recht offen, es werden jedoch Such- und Sortieralgorithmen, sowie Algorithmen zu Textanalysen genannt, unter die auch die Mustersuche in Texten fällt. Die Algorithmen sollen, nach der umgangssprachlichen Formulierung, auch implementiert werden.<sup>6</sup>

Somit ist die Themenwahl auch nach dem neuen Richtlinienentwurf begründet.

## **1.2 Verständnis des Themas**

Unter Verbesserungen von Algorithmen sollen hier solche verstanden werden, die nicht auf Veränderung von implementierungstechnischen Details, sondern auf neuen Ideen beruhen. Demzufolge soll hier auch nicht versucht werden, nach Entwicklung des naiven Verfahrens dieses zunächst zu optimieren, z.B. durch Vereinfachung von Schleifenbedingungen mit Hilfe eines Wächters als Kopie des Musters hinter dem Textende oder durch andere Maßnahmen.

Die Schüler sollen sich zunächst einen Überblick über das Themengebiet und die Problemstellung verschaffen, indem sie mögliche Anwendungsfälle sammeln und eine kurze Abgrenzung der Mustersuche in Texten zur Stichwortsuche in Verzeichnissen, etwa Telefonbüchern vornehmen.

Anschließend sollen das naive Suchverfahren BruteForce (BF)<sup>7</sup> sowie dessen Verbesserung QuickSearch (QS) entwickelt und implementiert werden. Dann soll noch mindestens eine mögliche Weiterentwicklung von QS, nämlich QuickSearch-Double (QS-D) in den Blick kommen.

Falls die Schüler während der Entwicklung von QS weitere Ideen äußern, können diese gegebenenfalls parallel zu QS-D mit QS gekoppelt und umgesetzt werden.

Der Dreischritt "Entwicklung eines (verbesserten) Verfahrens - Implementation - Test" soll hier jeweils für mindestens die drei genannten Verfahren durchlaufen

---

<sup>4</sup> vgl. Richtlinien S.34

<sup>5</sup> vgl. Neue Richtlinien S.13

<sup>6</sup> vgl. Neue Richtlinien S.26-27

<sup>7</sup> Für eine Beschreibung aller hier angesprochenen Verfahren s. Fachwissenschaftlicher Rahmen, S.6.

werden, wobei die Weiterentwicklung des jeweils vorangegangenen Algorithmus sich aus dessen Nachteilen motiviert. Ein großer vergleichender Test schließt sich an, um genauer festzustellen, was durch die Weiterentwicklung der Algorithmen denn nun erreicht wurde. Es soll einerseits bezüglich der Anzahl der Muster-Text-Vergleiche und andererseits bezüglich der Laufzeit verglichen werden.

Eine theoretische Laufzeitanalyse der Algorithmen ziehe ich für diesen Kurs nicht in Betracht: Die theoretische Analyse wäre für die Schüler dieses Grundkurses zu schwierig, da sie das O-Kalkül nicht kennen und eine Einführung darin besser anhand anderer Themengebiete, z.B. Sortierv Verfahren, gegeben werden kann. Außerdem geht es um einen praxisorientierten Vergleich der Verfahren im Mittel, wobei aber die average-case-Analyse der Mustersuche-Algorithmen zu schwierig ist. Daher ist in diesem Fall das systematische Testen ein angemessener Zugang zur Beurteilung der Algorithmen.

Die Reihe ließe sich noch ausbauen, indem noch weitere interessante Verfahren, gegebenenfalls mit völlig neuen Ansatz, wie etwa das Verfahren von Karb-Rabin (KR), das auf Hashing führt, behandelt werden könnten. Denkbar wäre z.B. in einem Leistungskurs die Behandlung von KR als Überleitung zu Hash-Funktionen.

Es bieten sich auch noch weitere Verfahren an, mit deren Hilfe man zu nachfolgenden Themengebieten überleiten könnte. Das Verfahren von Knuth-Morris-Pratt (KMP) könnte etwa hinführen zum Thema "Endliche Automaten",<sup>8</sup> oder die genauere Untersuchung verschiedener Implementationen ein und desselben Verfahrens könnte als Motivation dienen, sich intensiver mit den rechnerinternen Abläufen zu befassen, was zum Thema Assembler bzw. Hardware führen könnte.

Das Themengebiet Mustersuche ließe sich noch auf verwandte Problemstellungen, wie etwa Suchen regulärer Ausdrücke, Wortähnlichkeitsprüfungen oder Mustersuche mit vorgeschriebener maximaler Fehlerzahl (k-Mismatch) ausweiten, was den zeitlichen Rahmen allerdings stark ausdehnen und die einfache Problemstellung der Mustersuche erheblich verkomplizieren würde.

### **1.3 Planung und Darstellung der Stunden**

Im zweiten Kapitel wird die Planung der Unterrichtsreihe dargestellt. Die Auswahl der zu erarbeitenden Algorithmen wird vorgängig beschrieben, da sie eine zentrale Entscheidung darstellt, die die gesamte weitere Planung erheblich beeinflusst. Ebenso wird die Auswahl der verwendeten Medien gesondert begründet.

Keine der Entwicklungsphasen steht besonders im Vordergrund, wegen der geplanten Weiterentwicklung ist aber das Verständnis der jeweils vorangegangenen Verfahren von immenser Bedeutung, so daß der Entwicklung mindestens der ersten zwei Verfahren ein etwas größerer Stellenwert zugewiesen

---

<sup>8</sup> vgl. Sedgewick, S.284f.

wird. Die Planung, Durchführung und Auswertung des Tests sind integrale Bestandteile der Reihe. Allerdings kann die *Testdurchführung* kurz beschrieben werden, da es dort im wesentlichen um das Erstellen von Meßwerttabellen gehen wird.

Parallel zur zweiten Verbesserungsstufe können auch von Schülern vorgeschlagene Verfahren umgesetzt werden, so daß die Planung an dieser Stelle offen gestaltet sein wird, um Schülerinteressen hier angemessen berücksichtigen zu können.

## 2 Planung der Unterrichtsreihe

### 2.1 *Fachwissenschaftlicher Rahmen*

Da die Mustersuche nicht Standardschulstoff ist, möchte ich zunächst einen kurzen Überblick über das Fachgebiet und die von mir in Betracht gezogenen Verfahren geben.

Bei den einzelnen für diese Arbeit betrachteten Verfahren wird im folgenden die jeweilige Grundidee beschrieben, genauere algorithmische oder implementierungstechnische Details nur bei Bedarf. Weitergehende Beschreibungen finden sich in dem Überblicksartikel Hume / Sunday.

Im folgenden sind sowohl Text als auch Muster stets als *Array* gegeben, so daß auf jedes Zeichen frei zugegriffen werden kann. Ausgegeben werden soll nun die jeweilige Stelle, an der das Muster das erste Mal im Text auftaucht, oder die Meldung, daß das Muster nicht vorkommt.

#### ***Das naive Verfahren (BruteForce)***

Man legt das Muster unter den Textanfang und vergleicht zeichenweise von links nach rechts Musterzeichen und entsprechendes Textzeichen. Stimmen alle entsprechenden Zeichen überein, so ist das Muster gefunden; taucht ein Mismatch auf, d.h. stimmen ein Musterzeichen und das entsprechende Textzeichen nicht überein, so wird das Muster um eine Position nach rechts weitergeschoben, und die Musterzeichen werden wieder von links nach rechts mit den entsprechenden Textzeichen verglichen.

Beispiel: (Die Musterzeichen mit Kästchen werden mit den jeweils entsprechenden Textzeichen verglichen, die grau unterlegten Zeichen ergeben jeweils den Mismatch.)

Sie bewies wieder einmal, daß sie einiges konnte.

```
einige
  einige
    einige
      ...
        einige
          ...
```

einige

### Das QuickSearch-Verfahren (QS)

Grundlage hierfür ist BF. Taucht nun ein Mismatch auf, so soll um mehr als eine Position weitergeschoben werden. Dazu betrachtet man das erste Textzeichen direkt hinter dem Muster.

Taucht dieses Zeichen nicht im Muster auf, so kann das Muster ganz an diesem Zeichen vorbeigeschoben werden, ansonsten wird das Muster um soviel weitergeschoben, daß das letzte Auftauchen dieses Zeichens im Muster unter eben diesem Textzeichen liegt. Diese Shift-Werte hängen nur vom Muster ab und werden für jedes mögliche Textzeichen im voraus berechnet und in einem Array mit Indextyp Char gespeichert.<sup>9</sup>

Beispiel: (Die Textzeichen mit Pfeil werden jeweils als Index für die Shift-Tabelle benutzt.)

Er bewies dabei noch einmal, daß er einiges konnte.  
einige  
  einige  
    einige  
      einige  
        einige  
          einige  
            einige  
              einige  
                einige

Auszug aus der Shift-Tabelle:

Index	...	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{	...
Shift-Wert	...	7	7	1	7	2	7	3	7	7	7	7	4	7	7	7	7	7	7	7	7	7	7	7	7	7	...

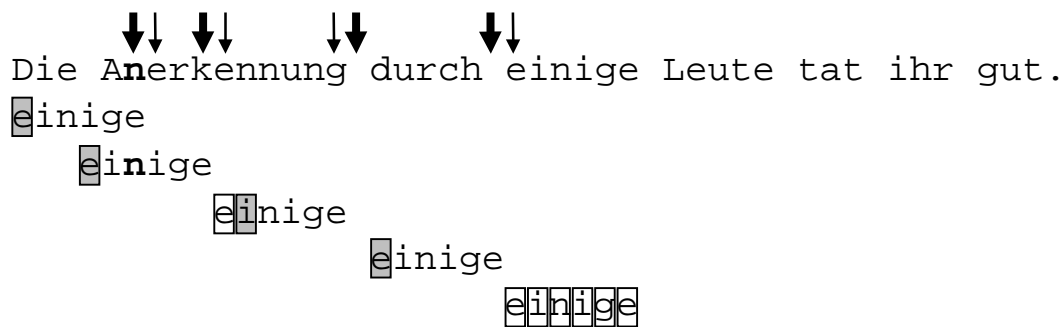
### Das QuickSearch-Double-Verfahren (QS-D)

Grundlage dieses Verfahrens ist QS, jedoch wird zusätzlich der um Eins verminderte Shift-Wert des Textzeichens direkt **über** dem Musterende betrachtet. Aus diesem Wert und dem Shift-Wert des darauffolgenden Textzeichens wird jeweils das Maximum gewählt.<sup>10</sup>

Beispiel: (Die beiden jeweils betrachteten Textzeichen sind mit Pfeilen markiert. Dasjenige mit günstigerem Shift-Wert ist mit dem dickeren Pfeil versehen.)

<sup>9</sup> s. Sunday, S.135

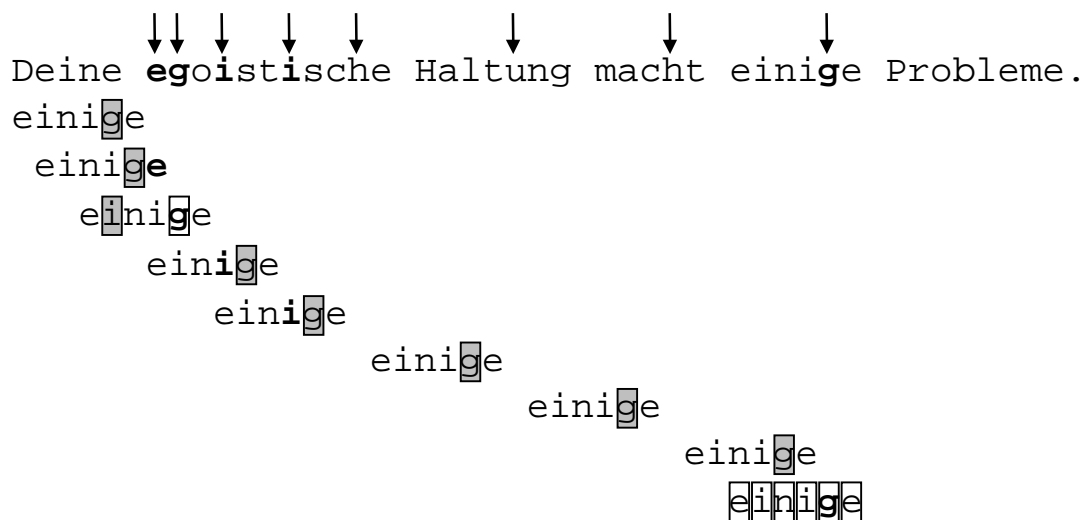
<sup>10</sup> s. Sunday, S.135 und Smith, S.1072


 Die Anerkennung durch einige Leute tat ihr gut.  
 einige  
   einige  
     einige  
       einige  
         einige

### **Das Optimal-Mismatch-Verfahren (OM)**

Die Grundlage dieses Verfahrens ist QS, aber es wird nicht von links nach rechts verglichen, sondern die Musterzeichen werden nach aufsteigender Auftretenshäufigkeit mit den jeweiligen Textzeichen verglichen, d.h. das Musterzeichen, das mit der geringsten Wahrscheinlichkeit im Text vorkommt, wird stets zuerst verglichen usw. Dabei ist die Verwendung von vorgegebenen festen Häufigkeitstabellen, gegebenenfalls angepaßt an Sprache und Texttyp, oder auch adaptive Anpassung an den jeweiligen Text während des Suchlaufs, z.B. über selbstsortierende lineare Listen, möglich.<sup>11</sup>

Beispiel: (Hier wird 'g' als der seltenste Buchstabe angenommen, gefolgt von 'i', 'n' und 'e'. In dieser Reihenfolge wird entsprechend verglichen.)


 Deine egoistische Haltung macht einige Probleme.  
 einige  
   einige  
     einige  
       einige  
         einige  
           einige  
             einige  
               einige  
                 einige  
                   einige

### **Das TurboSearch-Verfahren (TS)**

Grundlage hierfür ist QS. Es wird jedoch im Muster von rechts nach links verglichen.

<sup>11</sup> s. Sunday, S.136

Wird schon beim ersten Vergleich ein Mismatch zwischen dem letzten Musterzeichen und dem entsprechenden Textzeichen festgestellt, so arbeitet TS genau wie QS weiter. Findet sich aber ein Mismatch erst später, so wird dann die Kenntnis des Textzeichens über dem letzten Musterzeichen ausgenutzt: Kommt etwa das letzte Musterzeichen nicht noch einmal im Muster vor und schiebt man das Muster nach dem Mismatch "in Gedanken" um Musterlänge-1 Positionen weiter, so liegt das erste Musterzeichen unter dem bekannten Textzeichen, so daß sicher ein Mismatch auftreten wird. Man kann also ohne weitere Vergleiche wie in QS das Textzeichen hinter dem "in Gedanken" verschobenen Muster als Index für den Zugriff auf die Shift-Tabelle wählen. Im günstigsten Fall ergibt sich so ohne zusätzlichen Vergleich ein fast doppelt so weiter Sprung wie bei QS.

Kommt das letzte Musterzeichen noch einmal im Muster vor, kann "in Gedanken" entsprechend weniger weit verschoben werden.

Dieser einzelne "Gedanken"-Shift-Wert hängt nur vom Muster ab und wird ebenfalls in einer Vorverarbeitung ermittelt und in einer Integer-Variable gespeichert.

Beispiel: (Das "in Gedanken" verschobene Muster wird durchgestrichen notiert. Die wirklich verglichenen Musterzeichen erscheinen wieder in Kästchen. Der "Gedanken"-Shift-Wert beträgt beim diesem Muster 4, da zwischen dem letzten Musterzeichen 'e' und dem, vom Musterende aus gesehen, nächsten 'e' genau 4 Zeichen stehen.)

↓                    ↓                    ↓

Sie verwenden doch sicher einige meiner Beispiele.

einige

    einige

        einige

            einige

                einige

                    einige

                        einige

### **Das Verfahren von Knuth-Morris-Pratt (KMP)**

Grundlage ist hier BF. Tritt z.B. erst im j-ten Vergleich der Mismatch auf, so waren die ersten j-1 Vergleiche erfolgreich, d.h. dort stimmen Text und Muster überein, im nächsten Zeichen jedoch nicht. Aus diesem Wissen versucht man bei diesem Verfahren Vorteile zu ziehen. Treten etwa am Anfang des Musters keine sich wiederholenden Submuster auf, so wird man bei einem Mismatch an Position j um mehr als eine Position weiterschieben können.

Dazu berechnet man vom Anfangsstück des Musters mit Länge j-1 ein Endstück maximaler Länge k, das gleichzeitig ein Anfangsstück des gesamten

Musters ist. Dann wird das Muster so verschoben, daß die  $k+1$ -te Stelle im Muster unter dem aktuellen zuletzt verglichenen Textzeichen liegt, da dies die nächste Möglichkeit für eine Übereinstimmung ist. Dabei brauchen aber die  $k$  vorherigen Muster-Text-Vergleiche nicht ausgeführt zu werden, da die entsprechenden Zeichen nach Konstruktion von  $k$  übereinstimmen, so daß man zusätzlich noch Vergleiche einspart. Tritt wieder ein Mismatch auf, verfährt man entsprechend weiter.

Diese jeweiligen maximalen Längen werden für jede Position im Muster im voraus berechnet und in einem Array abgelegt.<sup>12</sup>

Das Verfahren hat den Vorteil, daß im Text nicht zurückgesprungen wird, d.h. kein Textzeichen wird mehrfach verglichen, was z.B. dann günstig ist, wenn sich nur ein Teil des Textes im Hauptspeicher befindet und die restlichen Teile etwa auf einem Magnetband.

Beispiel:<sup>13</sup> (In diesem Muster Hooligan taucht das H vom Anfang im ganzen Muster nicht mehr auf.)

Hoola-Hoola girls like Hooligans.

```
Hooligan
  Hooligan
    Hooligan
      Hooligan
        Hooligan
          ...
            Hooligan
```

### **Das Verfahren von Boyer-Moore (BM)**

Grundlage ist BF, es wird jedoch jeweils von rechts nach links verglichen.

Tritt dabei ein Mismatch auf, wird das Textzeichen direkt **über** dem Musterende betrachtet und ein Shift-Wert analog zu dem von QS berechnet (Vorkommensheuristik).<sup>14</sup> Zusätzlich können analog zu den Überlegungen in KMP aus den schon erfolgten Vergleichen Informationen gewonnen und genutzt werden (Match-Heuristik).

Beispiel:<sup>15</sup> (Der Deutlichkeit halber wird hier nur die Vorkommensheuristik verfolgt.)

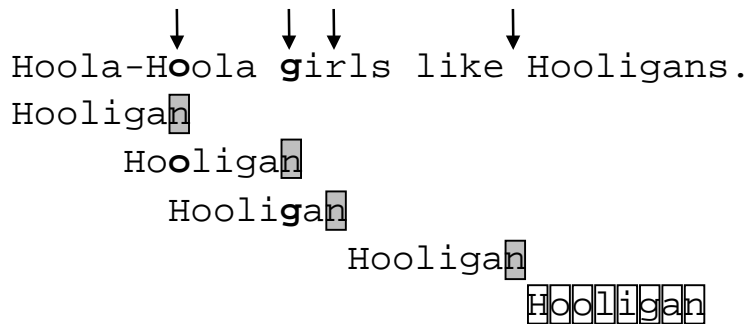
---

<sup>12</sup> s. Ottmann / Widmayer, S.630-634

<sup>13</sup> entnommen aus: Wirth, S. 62

<sup>14</sup> Diese Shift-Werte sind stets um 1 kleiner als die von QS, da ja dort das erste Textzeichen **hinter**, statt **über** dem Musterende betrachtet wird.

<sup>15</sup> entnommen aus: Wirth, S. 68



### **Das Verfahren von Karb-Rabin (KR)**

Dies Verfahren unterscheidet sich grundlegend von den bisherigen: Es handelt sich um einen probabilistischen Algorithmus.

Man faßt das Muster als Codierung einer Zahl im 255er-Zahlensystem auf, d.h. das Muster ABC würde der Zahl  $65 \cdot 255^2 + 66 \cdot 255 + 67$  entsprechen. Man wählt nun eine große Primzahl  $p$  und berechnet als Hashwert den Wert des Musters modulo  $p$ .

Man legt das Muster unter Textanfang und berechnet den Hashwert des entsprechenden Textabschnitts. Stimmt dieser mit dem Hashwert des Musters überein, so hat man mit hoher Wahrscheinlichkeit das Muster im Text gefunden.

Stimmen die Hashwerte nicht überein, so rückt man das Muster um eine Position weiter. Der Hashwert des nächsten Textabschnitts läßt sich leicht inkrementell aus dem des vorherigen Textabschnitts und dem neu hinzugekommenen Textzeichen berechnen.<sup>16</sup> Ein Beispiel würde hier nur Rechnungen wiedergeben und wird daher ausgelassen.

## **2.2 Intentionen der Unterrichtsreihe**

### **Sachspezifische Intentionen:**

Die Schüler

- gewinnen einen Einblick in ein modernes Themengebiet der Informatik: Sie kennen Anwendungsbeispiele der Mustersuche, erkennen die Bedeutsamkeit der Mustersuche und grenzen sie gegen andere Suchsituationen ab,
- entwickeln selbständig ein naives Verfahren zur Mustersuche,
- finden, simulieren, verbalisieren und implementieren Verbesserungen von Mustersuche-Verfahren,
- kennen die Vorverarbeitung, die Speicherung und Benutzung von Sprungweiten in einem `Array` als Hilfsmittel bei der Mustersuche,
- bewerten verschiedene Mustersuche-Verfahren und geben Gründe für unterschiedliches Verhalten bei Vergleichsanzahl und Laufzeit an.

<sup>16</sup> s. Ottmann / Widmayer, S.641-642

### **Fachspezifische Intentionen:**

Die Schüler

- koppeln unterschiedliche Ideen zur Verbesserung von Algorithmen,
- verfolgen Ideen zur Verbesserung von Algorithmen weiter,
- planen, konstruieren und führen einen Test durch zum Vergleich von Algorithmen,
- werten Testergebnisse zur Beurteilung von Algorithmen und zugrunde liegenden Ideen aus.

### **Soziale und affektive Intentionen:**

Die Schüler

- üben das Präsentieren von Arbeitsergebnissen: Sie stellen ihren Mitschülern ihre Ergebnisse mit Hilfe von Medien vor und machen sie ihnen verständlich,
- stellen ihre Arbeitsergebnisse zur Diskussion,
- akzeptieren Kritik an eigenen Lösungen und erkennen gegebenenfalls bessere Ergebnisse anderer Schüler als solche an,
- freuen sich über den erhaltenen Performance-Gewinn bei den Verfahren.

## **2.3 Lernvoraussetzungen**

### **Analyse der Lerngruppe**

Der Informatik-Grundkurs der 12. Jahrgangsstufe besteht aus sieben Jungen. Ich habe den Kurs in der Jahrgangsstufe 11 über 2 Monate unterrichtet, so daß mir die Schüler gut bekannt sind.

Das Kursklima ist sehr angenehm, alle Schüler verstehen sich untereinander gut, es gibt keine offensichtlichen Antipathien. Die Lern- und Arbeitsatmosphäre ist recht locker, wobei effektives und intensives Arbeiten möglich ist. Das Unterrichtsgespräch wird von den Schülern meist als Gelegenheit zum Diskutieren genutzt, wenn das Thema ihnen interessant und diskutierbar erscheint, wobei Jan und Markus K. manchmal die Gesprächsleitung zufällt. Es fühlen sich aber alle Schüler jederzeit frei, sich sowohl den Mitschülern als auch dem Lehrer gegenüber zu äußern.

Die Schüler arbeiten gut zusammen, so daß sich, auch wegen der geringen Kursstärke, oft aus Partnerarbeit ein Gespräch innerhalb der gesamten Lerngruppe ergibt, was zur Folge hat, daß verschiedene Gesprächs- und Arbeitsformen hier oft fließend ineinander übergehen. Wenn die Schüler Gelegenheit dazu haben, nehmen sie besonders in Gesprächssituationen oft auch das Unterrichtsgeschehen in die eigenen Hände, so daß für Gesprächsphasen bei dieser Lerngruppe, falls von der Sache her möglich, meist ein offener Impuls als Initiierung geeignet ist.

Die Schüler sind alle am Fach Informatik und auch an der Fachwissenschaft interessiert, was ich in Gesprächen mit einzelnen Schülern nach dem Unterricht, etwa über alternative Programmiersprachen, feststellen konnte.

Das Leistungsvermögen und die Leistungsbereitschaft sind als hoch anzusehen. Alle Schüler besitzen einen Computer und ein PASCAL-System und nehmen öfter Programmierarbeiten auf Diskette mit nach Hause oder bringen verbesserte Versionen von im Unterricht erstellten Programmen von zu Hause mit.

### **Organisatorische Rahmenbedingungen**

Die Unterrichtsstunden liegen montags in der ersten, donnerstags in der dritten und freitags in der sechsten Stunde, die nur 40 Minuten lang ist. Der Freitag eignet sich daher besonders für Phasen mit hoher Schüleraktivität und hohem Grad an Selbsttätigkeit, wie etwa Programmierarbeiten und Testdurchführungen.

In dem benutzten Informatikraum sind 15 Schülerrechner und 1 Lehrerrechner mit OHP-Display und ein OHP vorhanden. Eine entsprechende Projektionsfläche ist hinter der Tafel. Diese besteht aus zwei magnetischen, hintereinander angebrachten Einzelflächen. Sollen gleichzeitig Tafel und OHP eingesetzt werden, so muß eine andere Projektionsfläche, etwa durch eine umgedreht aufgehängte Landkarte bereitgestellt werden.

### **Einordnung der Unterrichtsreihe**

Der Kurs hat sich vor dieser Reihe mit einem Projekt "Strategiespiel" beschäftigt, in dessen Rahmen eine Mensch-vs.-Computer-Version des Spiels "Vier Gewinnt" entwickelt wurde.

Dabei war ein (2-dimensionales) `Array` die grundlegende Datenstruktur, so daß die Schüler im Umgang mit diesem Datentyp keine Schwierigkeiten haben dürften.

Zum Finden eines geeigneten Computerzugs wurde eine Strategie, die dem Mini-Max-Algorithmus entspricht, entwickelt. Ebenso werden auch bei der (Weiter-)Entwicklung von Verfahren zur Mustersuche Strategien benötigt, so daß auch hier ein Zusammenhang mit dem vorherigen Unterrichtsinhalt besteht.

Der Gegenstand der Unterrichtsreihe hat also, obwohl thematisch ohne direkten Bezug, fachmethodisch durchaus Zusammenhang mit den vorhergehenden Unterrichtsinhalten. Nach den Osterferien soll das Thema Hardware und Assembler behandelt werden, wobei sich die Motivation dafür eben aus dem Schluß dieser Reihe ergeben kann (s. Verständnis des Themas, S.4).

Die Frage, ob nicht die Mustersuche zusammen mit anderen Suchverfahren (lineare Suche, binäre Suche, Suche in komplexeren Datenstrukturen) zu behandeln und inhaltlich so systematisch gegen diese abzugrenzen wäre, möchte ich verneinen, da diese Abgrenzung weder für das eine noch die

anderen Themengebiete gewinnbringend wäre und die Tatsache, daß es sich um grundlegend verschiedene Problemstellungen handelt, den Schülern auch so klar ist.

## **2.4 Auswahl der Verfahren**

Der Zugang zum Thema über das naive BF-Verfahren ist die natürliche Vorgehensweise, so daß dieses als erstes Verfahren behandelt werden soll.

Die angestrebten Verbesserungen sollten dem jeweils vorangegangenen Algorithmus verwandte Versionen ergeben, einerseits damit die neue grundlegende Idee gegenüber dem jeweils vorhergehenden Verfahren deutlich hervortritt, andererseits damit beim abschließenden Test die Laufzeitunterschiede auch durch die jeweils zugrunde liegende Idee, und nicht durch etwaige Implementationsunterschiede (unterschiedliche Schleifentypen, unterschiedliche Abbruchbedingungen o.ä.) zu erklären sind.

Die erste Verbesserung sollte einen deutlichen Gewinn sowohl bezüglich der Laufzeit als auch bezüglich der Vergleichsanzahl gegenüber BF bringen, auch um noch weitere Verbesserungsversuche zu motivieren. Eine etwaige Vorverarbeitung sollte nicht zu schwierig sein, und das Verfahren sollte sich, ähnlich wie BF, als Basis für unterschiedliche Weiterentwicklungen eignen. Außerdem sollte es von den Schülern mit möglichst wenig Hilfestellung zu entwickeln und in unterschiedlichste Realisierungen von BF einzubauen sein.

Das Verfahren KMP ist zunächst auszuschließen: Die Grundüberlegung, das Wissen aus erfolgreichen Vergleichen zu nutzen, ist leicht zu finden, jedoch ist die Umsetzung für einen Grundkurs zu schwierig. Außerdem ist der Performancegewinn zu gering.<sup>17</sup>

Das BM-Verfahren ist in Betracht zu ziehen: Es wäre etwa möglich, zunächst die Vorkommensheuristik zu entwickeln und dann die Match-Heuristik zu erarbeiten. Diese Match-Heuristik ist jedoch recht aufwendig zu entwickeln und zu implementieren und trägt auch in der Praxis kaum etwas zur Laufzeitverbesserung bei, sondern verhindert nur eine schlechte worst-case-Laufzeit bei repetitiven Mustern.<sup>18</sup> Außerdem ist BM wegen der umgedrehten Vergleichsrichtung etwas schwierig aus dem naiven Verfahren herzuleiten.

Das QS-Verfahren ist einfacher und zudem schneller als BM<sup>19</sup> und eignet sich sehr gut als erste Verbesserung, da es alle o.g. Anforderungen gut erfüllt und zusätzlich später sehr leicht zu rekonstruieren ist, wenn man sich nur an BF und den Impuls "Betrachte das Textzeichen hinter dem Muster" erinnert. Außerdem kann die Shift-Tabelle bei der enaktiven Simulation zunächst außen vor blei-

---

<sup>17</sup> In der Praxis läuft KMP im Mittel nicht wesentlich schneller als BF, da oft schon beim ersten Vergleich ein Mismatch auftritt und so das Ausnutzen der erfolgreichen Vergleiche nicht stattfinden kann (s. Sunday, S.133). Das worst-case-Laufzeitverhalten von KMP ist allerdings erheblich besser als das von BF (s. Ottmann/Widmayer, S.634).

<sup>18</sup> vgl. Ottmann / Widmayer, S.641. Dieses Verfahren ist z.B. für eine Reihe günstig, in der eine theoretische Laufzeitanalyse durchgeführt werden soll.

<sup>19</sup> s. Sunday, S. 136

ben, da man auch durch scharfes Hinsehen erkennt, wie weit man das Muster nach einem Mismatch verschieben darf. Das QS-Verfahren bietet sich als Basis z.B. für QS-D, für OM (da die Reihenfolge, in der die Musterzeichen mit den jeweiligen Textzeichen verglichen werden, bei QS frei wählbar ist) oder auch für TS an, so daß hier auch je nach Schülerinteresse weiterverfahren werden kann.

Als eine mögliche zweite Weiterentwicklung möchte ich ein Verfahren wählen, das auch die Grenzen der QS-Heuristik "Weite Sprünge durch Vorausschau" aufzeigt. Es soll hier deutlich werden, daß man nicht mit jeder scheinbaren Weiterentwicklung auch eine Verbesserung im Sinne geringerer Laufzeit erreicht.<sup>20</sup> Andererseits sollte diese Erkenntnis nicht zu viel Zeit in Anspruch nehmen, so daß ein Verfahren mit leicht zu findender Idee und schneller Implementation gesucht ist. Das Verfahren QS-D erfüllt diese Anforderungen, da es die angegebene Heuristik weiterverfolgt und auch bezüglich der Vergleichsanzahl QS überlegen ist, jedoch bezüglich Laufzeit wegen der zusätzlichen Maximumsauswahl aus den zwei Shift-Werten unterliegt. Günstigerweise sieht man dies dem Verfahren nicht direkt an, so daß die Schüler auch auf den Test und dessen Ergebnisse neugierig werden. Eine Vorausberechnung der fraglichen Maxima mit Abspeicherung in einem quadratischen Array zur Einsparung der jedesmaligen Maximumsauswahl benötigt immens viel Speicherplatz und auch eine sehr lange Vorverarbeitungszeit für das Muster, so daß diese Maßnahme nicht weiterhelfen würde.

Sollte ein Schüler während der Reihe eine Idee für eine Verbesserung von QS haben, so kann diese, falls mit vernünftigem Zeitaufwand zu implementieren, von einem Teil der Lerngruppe parallel zu QS-D realisiert werden.

Die vollständige Entwicklung von OM ist zu schwierig und würde den Zeitrahmen der Reihe sprengen, so daß ich sie als zweite Erweiterung nicht in Betracht ziehe.

Wollte man die Heuristik "Weite Sprünge durch Vorausschau" weiterentwickeln, statt ihre Grenzen aufzuzeigen, wäre TS eine gute Wahl, da hier ja bei einem erfolgreichen Vergleich noch mehr Informationen für die Sprungweite gewonnen werden. Dies Verfahren ist allerdings etwas schwierig zu verstehen und auch langwierig zu implementieren.

Das KR-Verfahren stellt als probabilistisches Verfahren einen völlig anderen Zugang dar, so daß es weder als Verbesserung von BF noch als Verbesserung von QS in Frage kommt und von daher außen vor bleibt.

## **2.5 Einsatz von Medien**

In der Unterrichtsreihe sollen die zu erarbeitenden Verfahren von den Schülern entwickelt und den Mitschülern vorgestellt werden. Dazu wird ein gemeinsam zu betrachtendes und gemeinsam zu benutzendes Medium zur Präsentation bzw. zum Nachvollziehen und Korrigieren von Verfahren benötigt. Das Medi-

---

<sup>20</sup> vgl. Kapitel 1

um sollte die Dynamik der Verfahren widerspiegeln, d.h. das Muster sollte beweglich gegenüber dem Text sein.

Meine erste Idee waren Folien für den Text und Folienstücke für das jeweilige Muster. Jedoch wurde während der weiteren Planung klar, daß etwa zur Präsentation einer umgangssprachlichen Formulierung eines Verfahrens auf Folie und einer gleichzeitigen Durchführung des beschriebenen Verfahrens dann in mehreren Stunden zwei Projektoren benötigt würden. Dies erschien mir unangemessen. Außerdem ist das Hin- und Herschieben von Folienstücken eine recht filigrane Arbeit, bei der das Muster ab und an auch schon einmal schief oder zwischen zwei Textpositionen zu liegen kommt.

Daher entschied ich mich für die Tafel als Medium zur Simulation der Verfahren. Die Tafel im Informatikraum hat aber keine Karos, d.h. das jeweils korrekte Untereinanderschreiben des Musters unter den Text an der richtigen Position ist schwierig. Außerdem dauert das jedesmalige Neuschreiben des Musters nach der Verschiebung lange, so daß sich Simulationen an der Tafel unnötig in die Länge ziehen würden. Daher habe ich mich für Schilder entschieden, die sich mit Magnetplättchen an der Tafel befestigen lassen. Dabei steht das Muster auf fester Pappe, auf deren Rückseite Magneten geklebt sind, so daß es gut gehandhabt werden kann.<sup>21</sup>

Mit Hilfe der magnetischen Haftplättchen kann man jederzeit Muster oder Text von der Tafel abnehmen und wieder fixieren, oder man kann das Muster "mal kurz" unter eine andere Stelle des Textes heften usw.

Außerdem entfällt das Anschreiben von Muster und Text für den Lehrer während der Pausen, da die Modelle schon zu Hause vorbereitet werden können. Das Weiterschieben wird schön verdeutlicht, da man die Musterpappe auf der Tafel wirklich verschieben kann; außerdem kann man das Muster schnell mal gegen ein anderes austauschen, ohne lange schreiben zu müssen.

Sollte während der Reihe die Notwendigkeit entstehen, mal kurz ein Beispiel durchzuspielen, das direkt im Unterricht entstanden ist, so kann bei gleichbleibendem Text gegebenenfalls eine leere Musterpappe, die im Unterricht beschriftet wird, weiterhelfen. Soll nur der Text im Unterricht verändert werden, kann z.B. die vorhandene Musterpappe mit auf die Tafel geschriebenem Text kombiniert werden.

Die jeweils ausgewählten Muster sollten aus praktischen Gründen nicht zu lang sein, sollten aber einige Buchstaben mehrfach enthalten, damit etwa bei QS auch sicher das letzte Auftauchen des fraglichen Textzeichens im Muster betrachtet wird und klar wird, daß die anderen Stellen des Auftauchens uninteressant sind. Die Beispieltex te und Beispielmuster müssen also passend zur jeweiligen unterrichtlichen Situation geschickt gewählt werden.

Die Texte sollten dabei wegen der beschränkten Tafelbreite nicht zu lang sein, andererseits sollten damit alle möglichen Fälle abgedeckt werden, die in einem darauf anzuwendenden Verfahren vorkommen. Etwaige Spezialfälle soll-

ten aber erst im hinteren Textteil auftauchen. Die Beispiele müssen nach diesen Gesichtspunkten gewählt werden.

Als Beispieltexte und -muster für die Modelle werden statt der oft in Lehrbüchern vorkommenden abstrakten Ketten "abaaacdacc" richtige deutsche Sätze und Wörter verwendet. Dadurch wird den Schülern die Realitätsnähe der Problemstellung eher klar werden als bei offensichtlich konstruierten Beispielen.

Dieselben Texte und Muster wie beim Tafelmodell erhalten die Schüler jeweils als Tischmodell (s. S.67). Damit ist etwa Partnerarbeit für das selbständige Finden von Verfahren, bzw. für das genaue Beschreiben von Verfahren möglich. Da die Sozialform Partner- bzw. Gruppenarbeit in dieser Lerngruppe sehr effizient von den Schülern gestaltet wird, kommt diese mediale Ausstattung dem Kurs entgegen.

Tisch- und Tafelmodell sind bezüglich ihrer Funktionalität gleich, so daß die Schüler sich bei einem Phasenwechsel nicht jedesmal umstellen müssen. In beiden Modellen sind die Zeichen schon in `Array`-Kästchen geschrieben, die auch schon wie später in der PASCAL-Umsetzung durchnummeriert sind, um auch während der Programmierung gegebenenfalls auf die Modelle zurückgreifen zu können.

Diese Medien erlauben eine enaktive Darstellung der Verfahren, die schon aufgedruckte Numerierung der einzelnen Kästchen wie im `Array` erleichtert später den Übergang zur symbolischen Darstellung in PASCAL.

Zur Präsentation von Arbeitsergebnissen, etwa der umgangssprachlichen Beschreibung eines Verfahrens, sollen Folien dienen.

Für die Darstellung der Ideen, die den Verfahren zugrunde liegen, und deren Verknüpfung kann eine baumartige Darstellung an der Tafel hilfreich sein: Verschiedene Ideen werden als verschiedenfarbig markierte Knoten, gekoppelte bzw. zweifach angewandte Ideen als dann entsprechend doppelt markierte Knoten skizziert.<sup>22</sup> Das erleichtert auch das Sprechen über die Ideen bzw. über deren Verknüpfung.

Für die Programmierung der Suchverfahren wird mit einem Rahmenprogramm (s. S.69) eine Testumgebung zur Verfügung gestellt. Damit sind die Schüler von unnötiger Programmierarbeit befreit und können sich auf das Wesentliche konzentrieren. Die sogenannten nullterminierenden Strings darin werden verwendet, da für den abschließenden Test möglichst lange Texte (sehr viel länger als 255 Zeichen) benötigt werden und man auf normale `Array` vom Typ `Char` mit entsprechend großem Indexbereich nicht den `Readln`- oder `Writeln`-Befehl anwenden kann, was mit nullterminierenden Strings möglich ist. Diese werden ebenfalls als `Array` vom Typ `Char` deklariert, aber von Null an numeriert, so daß die Schüler lediglich ein wenig umdenken müssen.

---

<sup>21</sup> s. Foto auf Seite 33

<sup>22</sup> s. Tafelbild der 4. Stunde, S.71

In der Testphase wird dann diese Testumgebung u.a. durch den Einbau von Vergleichszählern, von Zeitmessungen und Einlesemöglichkeit des Textes aus einer Datei den Erfordernissen angepaßt, wobei zusätzlich benötigte Befehle in einer Unit bereitgestellt werden.

## **2.6 Begründungszusammenhang**

Die geplante Reihe soll 7 Stunden umfassen. Falls von den Schülern interessante und gewinnbringend erscheinende Vorschläge gemacht werden, denen sie noch nachgehen wollen, kann die Reihe auch auf 8 Stunden ausgedehnt werden, was allerdings den Nachteil hätte, daß die abschließende Reflexionsstunde auf den letzten Schultag vor den Osterferien in die sechste Stunde fallen würde.

Die Einführung in das Thema der Reihe soll nicht über ein spezielles Problem erfolgen, sondern ohne weitere Einschränkungen mit Hilfe eines informierenden Unterrichtseinstiegs. Dies begründet sich aus dem Interesse der Schüler auch an innerinformatischen Fragestellungen und an der Fachwissenschaft überhaupt, sowie aus der Tatsache, daß die Beschränkung bzw. Konzentration auf eine spezielle Anwendungssituation, wie etwa die Suche jedes Vorkommens eines Variablenbezeichners in einem Programmtext, weder sachspezifische noch motivationale Vorteile hätte, da die Schüler dieses Kurses sich gerade über die Vielfalt der Anwendungen, die sie auch zu Anfang der Reihe erarbeiten sollen, motivieren lassen.

Für das generelle Vorgehen in der Reihe sehe ich drei Möglichkeiten:

Eine Möglichkeit wäre, den Schülern alle zu erarbeitenden Verfahren z.B. in enaktiver oder ikonischer Form<sup>23</sup> vorzugeben und dann sprachlich formulieren und implementieren zu lassen. Dabei gehen aber sowohl das aktive Nachfinden der angestrebten Verfahren als auch der kreative Aspekt des Ausdenkens eigener Verfahren verloren. Außerdem wäre die **Verbesserung** der Verfahren recht unmotiviert.

Eine andere Möglichkeit wäre, die Schüler völlig ohne Hilfestellung zu Verbesserungen von BF aufzufordern. Es ist aber m.E. nicht davon auszugehen, daß die Schüler von alleine auf die so wichtige Idee der Vorausschau bzw. deren effiziente Umsetzung mit Hilfe eines `Array` kommen würden. Des weiteren scheint mir das Themengebiet nicht so direkt und ohne weiteres zugänglich zu sein, als daß gänzlich eigenständige Schülerarbeit vollkommen ohne Hilfestellung zu Anfang der Reihe schon möglich und sinnvoll wäre.

Ein geeigneter Kompromiß ist der hier gewählte Weg, die angestrebten Verfahren mit Hilfe von relativ offenen Impulsen zu initiieren und die Planung für weiterführende Schülervorschläge offen zu gestalten.

Die Entwicklung von Verbesserungen kann jeweils durch die Frage nach Nachteilen der im Moment vorhandenen Verfahren initiiert werden. Dabei

---

<sup>23</sup> Geeignet wäre etwa eine Darstellung wie auf Seite 6ff.

kann von den Schülern zunächst entweder nur ein Nachteil genannt, schon eine Verbesserungs**richtung** vorgeschlagen oder ein konkretes verbessertes Verfahren umrissen werden, so daß sich alle Schüler, unabhängig davon, wie weit sie jeweils schon gedacht haben, einbringen können.

Die Verfahren sollen von den Schülern jeweils in enaktiver, umgangssprachlicher und symbolischer, hier programmiersprachlicher, Form dargestellt werden, da sie dann einerseits sicherer zu verstehen und andererseits leichter zu behalten und zu rekonstruieren sind.

Die Möglichkeit, zuerst alle Verfahren bis zur enaktiven bzw. umgangssprachlichen Darstellungsebene zu entwickeln, um dann eine sehr lange zusammenhängende Programmierphase anzuschließen, habe ich verworfen, da die Lauffähigkeit der jeweiligen Implementation als Bestätigung der Richtigkeit der angestellten Überlegungen angesehen werden kann und durch das abwechselnde Entwickeln und Implementieren eine interessantere Phasierung entsteht. Außerdem sind die Sprech- und Denkweisen der Schüler in Bezug auf Algorithmen ohnehin so stark von PASCAL beeinflusst, daß sich Implementation und Test automatisch an die enaktive und umgangssprachliche Darstellung der Algorithmen anschließen. Ferner ist es für den Verlauf des Unterrichts wichtig, daß zu jedem Zeitpunkt die vorangegangenen Verfahren gut verstanden worden sind, um Verbesserungen anschließen zu können; dazu dient auch die jeweilige Implementation.

Das erste Vergleichskriterium, das die Schüler benutzen, wird die Anzahl benötigter Vergleiche sein, da diese Größe die einzige ist, die bei der Simulation der Verfahren in den Modellen zu messen ist.

Schon bei der Implementation von QS sollte sich bei den Schülern der Wunsch regen, auch zu testen, um wieviel **schneller** die Implementation des neuen Verfahrens denn tatsächlich ist. Spätestens nach der Entwicklung von QS-D stellt sich diese Frage ganz sicher, da die dort zusätzlich benötigte Maximumsauswahl ja auch Laufzeit kostet und die Frage, ob die im Mittel größeren Sprungweiten diese zusätzliche Operation aufwiegen. Diese Fragen motivieren den abschließenden Test.

Die nötigen Vorüberlegungen und Vorarbeiten sind in der Stunde vor dem Test zu leisten. Dazu werde ich eine Datei mit einem hinreichend langen Text zur Verfügung stellen. Die Testmodalitäten sowie die daraus resultierende Vorbereitungsarbeit sollen vorher mit den Schülern zusammen erarbeitet werden.

Wegen der großen Ideenvielfalt des Themengebiets können sich fast in jeder Phase Diskussionen z.B. über neue Ideen für Verbesserungen oder über Situationen, in denen ein Verfahren besonders gut oder schlecht greift, ergeben, die dann die Planung entsprechend beeinflussen können. Darauf werde ich gegebenenfalls an den entsprechenden Stellen eingehen.

Es ergibt sich insgesamt daher folgende Grobplanung:

Stunde	Inhalt	Kommentar / Medien / mögl. HA
1	Themenbekanntgabe, Sammlung von Einsatzgebieten, Abgrenzung von Texten ohne zusätzliche Struktur (Mustersuche) gegenüber Texten mit zusätzlicher Struktur (Stichwortsuche), Umgangssprachliche Formulierung von BF, Simulation am Modell	Unterscheidung Wort vs. Muster Einführung der Modelle (Tafel- und Tischmodell) HA: Analyse des Rahmenprogramms , PASCAL-Prozedur zu BF
2	Implementierung BF, Erarbeitung von QS auf enaktiver Eben, jeweils mit Begründung für erlaubte Sprungweite und Erläuterung der Bezeichnung "Vorausschauender Algorithmus"	Modelle HA: Umgangssprachliche Formulierung von QS
3	Vorbereitungen zur Programmierung von QS. Falls Zeit bleibt: Beginn der Implementierung von QS	HA: Restliche Implementierung
4	Falls im bisherigen Verlauf der Reihe keine Mustersuche-Ideen von S. geäußert wurden, kann hier eine Art Ideensammlung stattfinden, die ich gegebenenfalls mit vorgegebenen Muster-Text-Kombinationen, die auf eine bestimmte Verbesserungsrichtung hindeuten, initiieren kann. Parallele Entwicklung und Implementierung von QS-D und einer Kopplung eines Schülervorschlags mit QS in Gruppen	Ideenbaum, Modelle
5	Testplanung und Durchführung der nötigen Testvorbereitung (Zähler einbauen, Zeitmessungen einbauen, Text-Datei einlesen usw.)	HA: Muster gemäß Anforderungen für den Test ausdenken
6	Testdurchführung, Notation der Meßergebnisse auf vorbereitete Arbeitsblätter	L. bereitet Graphiken für Testauswertung vor.
7	Testauswertung, Bewertung der Verfahren und der zugrunde liegenden Ideen, Rückblick und Ausblick	

### 3 Die erste Stunde

#### 3.1 Planung

##### Thema der Stunde:

Einsatzfelder der Mustersuche, Abgrenzung der Mustersuche gegenüber Suche in Texten mit zusätzlicher Suchstruktur sowie Entwicklung eines einfachen Verfahrens zur Mustersuche (BF)

##### Intentionen:

Die Schüler sollen die Problemstellung bei der Mustersuche sowie verschiedene Anwendungsbeispiele kennen, die Bedeutsamkeit der Mustersuche erfassen und die Unterscheidung zwischen Texten ohne und mit zusätzlicher Suchstruktur treffen.

Sie sollen ein einfaches Verfahren zur Mustersuche auf enaktiver Ebene entwickeln und es mit Hilfe einer umgangssprachlicher Beschreibung und einer Simulation präzisieren und fixieren.

Sie legen dabei ihre Arbeitsergebnisse den Mitschülern vor, stellen diese zur Diskussion und korrigieren gemeinsam mit ihren Mitschülern gegebenenfalls ihre Lösung.

In der Hausarbeit analysieren sie das Rahmenprogramm und entwickeln selbstständig aus der umgangssprachlichen Beschreibung eine dazu passende PASCAL-Funktion.

### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Informierender Einstieg	L. schreibt das Thema der Unterrichtsreihe an die Tafel, präsentiert das Modell und gibt einen kurzen Überblick über die Reihe: "Wir werden uns für die nächste Zeit mit Mustersuche in Texten beschäftigen, zunächst ein einfaches Verfahren entwickeln, uns dann um Verbesserungen bemühen und diese schließlich genauer beurteilen". Ein S. zeigt am Modell, wo das gegebene Muster gefunden wird. "Bevor wir in die Algorithmik einsteigen, solltet ihr euch einen gewissen Überblick über das Themengebiet verschaffen, indem ihr Anwendungsfälle überlegt und diese Suchsituationen noch gegen gewisse andere abgrenzt."	Tafel Tafelmodell	LV  S.-Aktion
2) Erarbeitung I a)	S. erarbeiten Anwendungsfälle dieser Mustersuche.	AB 1	PA
	b) Anwendungsbeispiele werden an der Tafel gesammelt und kurz erläutert (Was ist Text, was Muster?).	Tafel	UG
	c) L. fordert je einen S. auf, im Lexikon die Namen von zwei "bekannten Informatikern", nämlich Lange und Hinrichs bzw. in einem EDV-Buch das Muster "Lautsprecher" zu suchen.	Lexikon EDV-Buch	S.-Aktion
3) Sicherung I	S. erläutern, was sie sich beim Lexikon bzw. EDV-Buch zunutze gemacht haben (alphabetische Sortierung der Stichwörter bzw. Indizierung), und die Unterscheidung zwischen "Text ohne zusätzliche Suchstruktur" und "Text mit zusätzlicher Suchstruktur" wird erarbeitet und deren Bedeutung für den Suchvorgang geklärt.	Tafel	UG

4) Erarbeitung II	S. überlegen ein einfaches Verfahren zur Muster- suche und fixieren eine umgangssprachliche, aber genaue Formulierung des Vorgehens. Eine Gruppe schreibt dabei auf Folie.	Tischmodell AB 2 (s. S.68)  Folie	PA
5) Sicherung II	Die S.-Gruppe legt ihre Folie auf, ein <b>anderer</b> S. führt das Verfahren anhand der Beschreibung an der Tafel zuerst mit einem vorkommenden, dann mit einem nicht vorkommenden Muster vor. Fehlende oder unrichtige Teile werden ergänzt bzw. korrigiert.	Folie Modell	S.-Aktion
Hausaufgabe	L. händigt S. den Rahmenprogrammtext aus mit dem Hinweis auf die Indizierung der Array-Typen für Text und Muster von Null an. Sie sollen das kurze Programm analysieren und zu Hause eine passende PASCAL-Funktion schreiben, die das Muster im Text sucht und eine entsprechende Meldung zurückgibt. Zwei S. erhalten eine Folie, um ihre HA in der nächsten Stunde vorstellen zu können.	HA-Folien	EA

### Didaktisch-methodische Diskussion:

Auf den informierenden Einstieg bin ich schon im Begründungszusammenhang eingegangen. Die frühe Präsentation des Modells dient einer ungefähren Vorstellung der Problemstellung, die durch die Beispielsammlung und die spätere Simulation präzisiert wird. Das wahrscheinlich auftretende Mißverständnis, daß Leerzeichen eine andere Rolle als die anderen Zeichen spielen, soll gleich zu Anfang durch das entsprechend gewählte Beispiel ausgeräumt werden: Das gesuchte Muster taucht zwar im Text als Wort auf, ist aber vorher schon einmal Teil eines längeren Wortes (s. S.67).

Eine Formalisierung der Problemstellung in einer mathematischen Formulierung ist hier unnötig.

Die Beispielsuche soll einerseits den Schülern die Bedeutsamkeit des Themas vor Augen führen, da (hoffentlich) Beispiele aus verschiedenen Bereichen der Informationsverarbeitung angeführt werden. Andererseits kann sie, wenn falsche Beispiele genannt werden, zu einer genauen Klärung der Problemstellung sowie verwandter und auch gegensätzlicher Aufgabenklassen beitragen. Auf dem dazu ausgeteilten Arbeitsblatt 1 sind schon zwei Beispiele notiert (s. Tafelbild, S.66), um die Phantasie anzuregen, ohne aber gleich die gängigsten Beispiele vorwegzunehmen. Die Sammlung an der Tafel dient auch dazu, korrekte Beispiele von falschen zu trennen und die Begriffe Text und Muster zu festigen.

Das Präsentieren der völlig anderen Suchsituationen "Lexikon" und "EDV-Buch mit Index" schließt sich logisch gut an die Beispielsammlung für Mustersuche an. Die Schüler wissen natürlich, wie man in einem Lexikon und in einem Index sucht, um aber die Unterscheidung zwischen Texten mit und Texten ohne Suchstruktur deutlicher hervortreten zu lassen, will ich je einen Schüler die entsprechende Suche durchführen lassen. Zur Auflockerung lasse ich die Schüler nach den Namen zweier "berühmter Informatiker" suchen, wobei ich meinen Nachnamen und den des Fachlehrers gewählt habe. Dabei tauchen beide Personen nicht im Lexikon auf, das gesuchte Wort "Lautsprecher" im Index des EDV-Buchs schon, so daß auch hier insgesamt beide Fälle des Vorkommens und Nichtvorkommens des Suchelements auftreten. Die Alternative, im Informatik-Duden oder einem Fachbuch etwa das Muster "Mustersuche" finden und die Erklärungen vorlesen zu lassen, habe ich abgelehnt, da diese dort meist sehr weitschweifig und formal sind, so daß das Vorlesen nur unnötig viel Zeit gekostet hätte, ohne der Sache an dieser Stelle dienlich zu sein.

Ich gehe davon aus, daß die Schüler bei der umgangssprachlichen Formulierung das Abbruchkriterium für nicht vorkommende Muster zunächst nicht einbauen werden, so daß ich bewußt als erstes ein Muster gewählt habe, das auch im Text vorkommt, um die Schüler erleben zu lassen, wie nachlässig man oft Verfahrensvorschriften formuliert. Das zweite Muster kommt nicht im Text vor, damit die Schüler die Folienversion und ihre eigenen Versionen korrigieren. Daher soll auch ein Schüler simulieren, der nicht an der Folie mitgearbeitet hat. Die Korrektur ergibt sich direkt aus der zweiten Simulation.

Die umgangssprachliche Formulierung ist hier besonders griffig, sie dient der Fixierung der Arbeitsergebnisse und kann Basis für die entsprechende PASCAL-Funktion sein, so daß die Hausaufgabe für alle sinnvoll und lösbar ist.

### **3.2 Durchführung**

Der Unterricht verläuft bis zur Phase 2a wie geplant. Der Lehrer gibt den vorgesehenen Überblick, und ein Schüler heftet die mit Magneten hinterklebte Pappe mit dem Muster zunächst an die Stelle weiter hinten im Text, wo das Muster als ganzes Wort auftaucht, korrigiert sich dann ohne weiteres Zutun des Lehrers selbst und erklärt nach der Frage, was er denn im Kopf gehabt habe, den Unterschied zwischen Muster und Wort. Damit ist ein grobes Verständnis der Problemstellung gegeben, die in der folgenden Phase präzisiert wird:

Die Erarbeitung von Beispielen verläuft in Partnerarbeit, bei der Sammlung an der Tafel zeigt sich, daß die Schüler viele gute Ideen haben (s. Tafelbild, S.66). Die Beispiele werden jeweils auf die Frage des Lehrer, was hier Muster, was Text sei, näher erläutert, und unrichtige Beispiele, wie etwa die zunächst genannte Rechtschreibprüfung, werden nach dieser näheren Erläuterung von den Schülern wieder verworfen. Insgesamt zeigt sich ein sehr weites Verständnis von Mustersuche, was etwa die Beispiele Spracherkennung und Bild-

vergleich zeigen. Diese Phase nimmt wesentlich mehr Zeit in Anspruch als geplant, da die Schüler einerseits viele Beispiele präsentieren, diese aber im Gespräch auch kritisch beleuchten. Der Lehrer entschließt sich, diese Phase auszudehnen, da sie produktiv scheint, die Beispiele interessant sind und von den Schülern akribisch genug, aber nicht zu detailliert bedacht und erläutert werden.

Als Dirk den - vom Lehrer selbst vorher nicht gesehenen - Zusammenhang zwischen dem Projektthema "Vier gewinnt" durch das Beispiel "Finden eines 'Dreiers' als Muster im Spielfeld als Text" herstellt, sind die meisten Schüler und der Lehrer erstaunt und Stefan verzieht das Gesicht: "Warum haben wir das denn dann nicht vorher gemacht?" Dirk erläutert noch den Unterschied zwischen der Mustersuche wie im Tafelmodell und der Mustersuche im Vier-gewinnt-Spielfeld ("Das ist dasselbe, nur zweidimensional!").

Da schon viele gute Beispiele notiert sind, werden am Schluß der Sammlungsphase 2b die Beispiele von den Schülern nur noch genannt und erläutert, sie werden jedoch nicht mehr an der Tafel fixiert.

Phase 2c verläuft dann sehr zügig mit dem Hinweis eines Schülers, daß entweder das Lexikon alt oder die beiden "berühmten Informatiker" wohl doch nicht so berühmt seien, da es wohl den Namen Lange, aber keinen Eintrag G. Lange gebe und überhaupt keinen Hinweis auf den Namen Hinrichs. Der Schüler mit dem Buch mit Index nimmt diesen sofort zur Hilfe und findet das gesuchte Wort ohne weiteres.

Bei der Frage nach dem Unterschied zwischen den notierten Beispielsituationen der Mustersuche und dem Suchen im Buch mit Index oder Lexikon bemerkt Jan treffend, daß dort die Mustersuche quasi "schon eingebaut sei", was in den Tafelanschrieb "Texte mit Suchstruktur" mündet. Derselbe Schüler fordert als Zusammenfassung für die Mustersuchsituationen auf der linken Tafelseite dann den auch recht passenden Ausdruck "Rohtexte". Dieser wird zusätzlich zu dem von Eike genannten Begriff "Texte mit Suchstruktur" an der Tafel notiert.

Die Erarbeitung der umgangssprachlichen Beschreibung verläuft ebenfalls recht zügig, drei der vier Gruppen erarbeiten den BF-Algorithmus innerhalb weniger Minuten, alle jedoch ohne das Abbruchkriterium bei erfolgloser Suche.

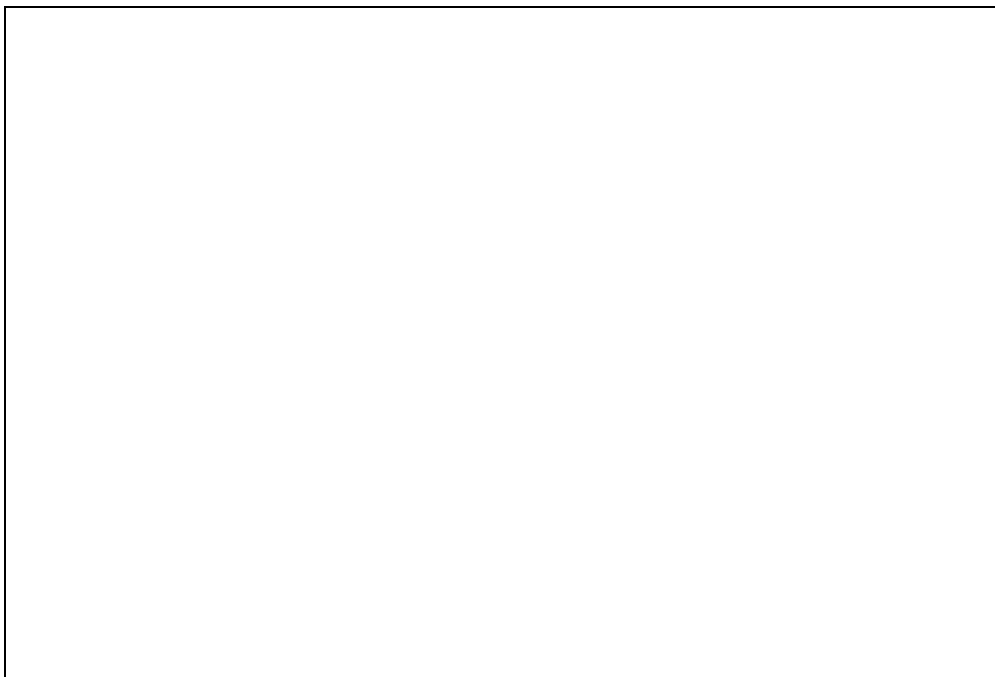
Jan legt die mit Markus K. zusammen erarbeitete Folie auf und setzt sich etwas konsterniert nach der Aufforderung des Lehrers, ein anderer Schüler solle das Verfahren durchführen, wieder hin.

Da die Projektionsfläche von der Tafel aus nicht eingesehen werden kann, liest Björn die jeweiligen Anweisungen vor, die Stefan ausführt.

Die Gruppe Markus R. / Björn schlägt nun ein schon verbessertes Verfahren vor: Man sucht sich den Buchstaben im Muster, der die geringste Auftretenshäufigkeit hat, und sucht diesen im Text. Dann legt man das Muster passend darunter und vergleicht gegebenenfalls die restlichen Buchstaben usw.

Dies wird vom Lehrer entsprechend gewürdigt, jedoch mit dem Hinweis zurückgestellt, daß es zunächst noch um ein einfaches Verfahren ging, aber diese Idee später noch mal aufgenommen werden würde.<sup>24</sup>

Bei der Bitte nach einer zweiten Simulation mit einem anderen Muster geht Eike nach vorne und beginnt, stockt aber an der Stelle, wo das Muster "seinem" unter das Wort "Sie" zu liegen kommt, und fragt, ob die Suche auf Groß- und Kleinschreibung achte, was allgemein bejaht wird. Da der erste Buchstabe des Musters (sonst) nicht vorkommt, "spult" er vor, bis das Musterende kurz vor dem Textende liegt, und bemerkt richtig, daß das Verfahren abbrechen könne, sobald das Muster nicht mehr "hinpasse", was auf der Folie mit Hilfe des Halbsatzes "(...) oder das Musterende über das Textende hinausragt." farblich abgesetzt ergänzt wird (s. S.68). Die übrigen Gruppen ergänzen dies analog in ihren Versionen auf ihren Arbeitsblättern.



Eike nimmt die nötige Ergänzung der Beschreibung vor.

Als der Lehrer die Hausaufgabe gestellt hat, klingelt es zur Pause. Es bleibt keine Zeit für eine abschließende Zusammenfassung, aber da die Anteile am Unterrichtsgeschehen recht gleichmäßig auf alle Schüler verteilt waren und die wichtigsten Ergebnisse einerseits im Tafelbild (Anwendungsfälle und Abgrenzung), andererseits auf den Arbeitsblättern (Umgangssprachliche Formulierung) fixiert sind, ist dies auch nicht nötig.

### **3.3 Reflexion**

Der informierende Unterrichtseinstieg war einerseits von der Sache her angemessen (s. didaktisch-methodische Diskussion), andererseits war er hier schü-

---

<sup>24</sup> Dies ist die grundlegende Idee des Verfahrens Optimal Mismatch (s. Fachwissenschaftlicher Rahmen, S.6), jedoch beschränkt auf denjenigen Musterbuchstaben mit der geringsten Auftretenshäufigkeit im Text.

lergerecht, was die Aufmerksamkeit und das Interesse der Schüler während der gesamten Stunde, vor allem aber während der Beispielsammlung, in der alle mindestens ein Beispiel präsentiert haben, gezeigt hat. Die interessante Idee von Markus R. und Björn, nicht das erste Musterzeichen, sondern das seltenste als erstes zu vergleichen, zeugte ebenfalls von regem Interesse am Thema.

Das sehr weite Verständnis von Mustersuche wurde durch das recht frühzeitig von mir präsentierte, stark einschränkende Modell nicht gestört, was auch die von Schülern hergestellten Zusammenhänge, wie der zum vorhergehenden Projekt "Vier gewinnt", zeigten.

Die teilweise unpassenden bzw. zunächst nicht gut durchdachten Beispiele zeigten, wie wichtig eine gemeinsame Sammlung und Erläuterung war, hierfür war die jeweilige Frage "Was ist in deinem Beispiel das Muster, was der Text?" gut geeignet. Die Schüler haben gehaltvoll begründet, warum es sich um ein Beispiel für Mustersuche handelt oder warum nicht.

Die Kontrastsituationen "Lexikon" und "Buch mit Index" kamen gut an, die Entscheidung, kein Informatik-Lexikon oder Fachbuch über Mustersuche zu nehmen und das Stichwort "Mustersuche" oder ähnliches suchen und die Erläuterungen vorlesen zu lassen, war im Hinblick auf die knappe Zeit richtig, da die Schüler auch so durch die vielen von ihnen erläuterten Beispiele eine richtige Vorstellung von der Problemstellung hatten.

Die Zeit für die Beispielsammlung war gut investiert, da die Schüler sich dadurch klarer über die Problemstellung waren. Dies zeigte sich auch darin, wie zügig die Beschreibung des einfachen Verfahrens und die Simulation dieses Verfahrens am Schluß der Stunde vonstatten gingen.

Das Tischmodell hat sich bewährt, was sich in der schnellen Erarbeitung von BF zeigt: Hier hatte jede Schülergruppe ein gemeinsames Anschauungsmittel zur Verfügung, und außerdem wurden durch das Tischmodell auch Begriffe für die umgangssprachliche Beschreibung angeregt (s. "weitergeschoben" und "hinausragt" auf der Folie, S.68).

Das Tafelmodell hat sich ebenfalls gut bewährt. Die Simulation des auf der Folie beschriebenen Verfahrens wurde zu Anfang sehr akribisch, bei unkritischen, bzw. uninteressanten Passagen im "Schnelldurchlauf" durchgeführt, was durch das schnelle Schieben des Musters gut deutlich wurde. Hier zeigt sich auch der Vorteil dieses Modells gegenüber dem wiederholten Hinschreiben des Musters. Die Tatsache, daß sich sofort ein Schüler für die Simulation an der Tafel gemeldet hat, zeigt, daß das Modell auch optisch und haptisch ansprechend ist.

Die Beispieltexte und -muster für die Modelle waren angemessen: Dadurch, daß im ersten Beispiel das Muster auftaucht, haben sich die Schüler zunächst voll auf das Grundgerüst des Verfahrens ohne zweites Abbruchkriterium konzentriert und dieses dann beim zweiten Beispiel entsprechend nachgetragen. Das zweite Muster war insofern günstig gewählt, als daß gut "weitergespult"

werden konnte, da schon der erste Musterbuchstabe im Text gar nicht vorkam. Trotzdem konnte hier als Nebenprodukt die Frage nach Unterscheidung zwischen Groß- und Kleinschreibung schnell geklärt werden.

Insgesamt bin ich mit der Stunde äußerst zufrieden:

Das Thema hat die Schüler angesprochen, was die große Fülle von Beispielen ebenso zeigt wie die teilweise unter den Schülern aufkommenden Diskussionen, ob ein Anwendungsfeld denn nun ein Beispiel für Mustersuche beinhalte oder nicht, vor allem beim Beispiel Internet. Die Abgrenzung gegen andere Suchsituationen ist gut vollzogen worden, was auch die Erklärung von Jan ("Da ist die Mustersuche schon eingebaut") zeigt. Da der BF-Algorithmus von allen gefunden wurde, kann in der darauffolgenden Stunde mit dem QS-Verfahren begonnen werden.

Die Simulation an der Tafel war sinnvoll, da hier das Fehlen des Abbruchkriteriums erkannt und die umgangssprachliche Formulierung auf der Folie entsprechend korrigiert wurde; die Schüler haben die Simulation auch gerne durchgeführt, die Methode "Simulation" kann also bei den nachfolgenden Verfahren wahrscheinlich wieder mit Gewinn eingesetzt werden.

Die Tatsache, daß schon in einer so frühen Phase der Reihe ein Verbesserungsvorschlag für BF gemacht wurde, ohne daß dieses gefordert war, zeigt deutlich, daß die Verbesserung von Algorithmen für die Schüler interessant und bedeutsam ist. Die sich sachlogisch ergebende Kategorisierung der Mustersuche-Verfahren (Bereich "Schnelles Finden des Mismatch" vs. Bereich "Schnelles Weiterschieben des Musters")<sup>25</sup> bietet sich schon an dieser Stelle an, so daß sie in der zweiten Stunde als Binärbaum an der Tafel stehen soll, damit die Schüler den Standort der eigenen Ideen und der neu zu entwickelnden Verfahren erkennen.

Weiterhin hat diese frühe Schüleridee insofern Auswirkungen auf die Grobplanung, als daß in der vierten Stunde keine von mir initiierte Ideensammlung mehr stattzufinden braucht, da einerseits in dieser Stunde alle Schüler von der Idee von Markus R. und Björn angetan waren und diese andererseits einen guten Kandidaten zur Implementation und für weitere Betrachtungen (Wie kommt man an die benötigten relativen Häufigkeiten? Ist das Verfahren *immer* besser? usw.) darstellt. Statt dessen möchte ich die Schüler in der vierten Stunde in Neigungsgruppen einteilen: Die eine Gruppe spürt der Idee der "Vorausschau" von QS nach, entwickelt QS-D und stellt am Schluß das Ergebnis den andern vor. Die zweite Gruppe koppelt die Idee von Markus R. und Björn, die ich als SB (Seltenster Buchstabe) bezeichnen möchte, mit QS zu QS-SB. Eine vollständige Version von OM wäre hier zuviel, aber die "Light"-Version, in der nur dasjenige Musterzeichen mit der niedrigsten Vorkommenshäufigkeit von Hand verglichen wird, ist in angemessener Zeit zu entwickeln und zu implementieren. Eine Sammlung von weiteren Ideen und Verfahren der Schüler kann in der letzten Stunde als Ausblick erfolgen.

Die Vertröstung von Markus R. und Björn auf später bleibt als kleiner Wermutstropfen. Ich halte es aber weder für sinnvoll, als erste Verbesserung von BF das reine SB-Verfahren zu thematisieren, da es in der PASCAL-Version gar nicht wesentlich schneller ist als BF, noch halte ich es für angemessen schon in dieser Phase mehrere Algorithmen parallel entwickeln zu lassen. Außerdem ist die Vorverarbeitung des Musters und das schnelle Weiterschieben im Bereich Mustersuche von grundlegenderer Bedeutung als die Idee von SB. Insgesamt möchte ich daher an der Idee festhalten, zunächst QS zu entwickeln, plane aber, in der nächsten Stunde die beiden Schlüsselideen ("Schnelles Finden des Mismatch" und "Sprunghaftes Weiterschieben des Musters ") zu thematisieren und die Aufmerksamkeit offen auf die zweite Verbesserungsrichtung zu lenken und die Betrachtung der ersten Verbesserungsrichtung auf die vierte Stunde zu vertagen. Die Idee wurde so pointiert vorgetragen und ist auch so einfach, daß sich sicher auch in späteren Stunden alle daran erinnern können.

## **4 Die zweite Stunde**

### **4.1 Planung**

#### **Thema der Stunde:**

Implementation und Test des naiven Verfahrens (BruteForce) zur Mustersuche, Entwicklung einer ersten Verbesserung (QuickSearch) auf enaktiver Ebene, Einordnung der Verbesserungsrichtungen in einen Ideenbaum<sup>26</sup>

#### **Intentionen:**

Die Schüler sollen ihre zu Hause entwickelte PASCAL-Prozedur/-Funktion zum naiven Verfahren den Mitschülern vorstellen, zur Lösung ihrer Mitschüler Stellung beziehen und ihre jeweilige Lösung in das Rahmenprogramm einbauen und testen.

Die Schüler sollen Mängel von BF erkennen und mindestens zwei Verbesserungsrichtungen benennen. Sie sollen die Idee von QS finden, das Verfahren in einer Simulation durchführen und ihren Mitschülern dabei erläutern und verständlich machen sowie die Entstehung der Sprungweiten an Beispielen begründen.

---

<sup>25</sup> Die Trennung dieser beiden Verbesserungsstrategien findet sich so in Gronek, S.279-280.

<sup>26</sup> Die vollständige Version dieses Baums findet sich im Tafelbild der 4.Stunde (S.71). In dieser Stunde werden nur die ersten beiden Ebenen entwickelt.

In der Hausaufgabe sollen sie auf der Grundlage ihrer umgangssprachlichen Formulierungen des BF-Verfahrens eine umgangssprachliche Formulierung des Verfahrens QS liefern.

### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Anknüpfung an vorherige Stunde	Ein S. gibt einen kurzen Überblick über den bisherigen Verlauf der Reihe. Ein S. stellt seine Lösung der HA vor, Kritik kann ggf. wieder am Modell deutlich gemacht werden, dazu sind zwei Haftschilder, auf die Variablenbezeichner geschrieben werden können, vorbereitet.	HA-Folie Modell mit zus. Schildern	SV
2) Einbau in das Rahmenprogramm	S. bauen ihre PASCAL-Prozedur in das Rahmenprogramm, das über das Netz zur Verfügung gestellt wird, ein und testen es.	Computer	EA
3) Problemaufwurf / Erarbeitung	L. fragt nach Nachteilen von BF und möglichen Strategien für die Verbesserung. Ein S. macht die Idee von SB noch mal klar und erläutert den dadurch erreichten Effekt ("Schnelles Finden des Mismatch"). Dies wird als linker Sohn des Wurzelknotens BF fixiert. Als zweiter Nachteil von BF sollte noch erkannt werden, daß dabei immer nur um <b>eine</b> Position weitergeschoben wird. L. lenkt mit Hinweis auf spätere Betrachtung der ersten Strategie die Aufmerksamkeit auf die zweite Verbesserungsrichtung.	Tafelmodell mit Muster "enzymatisch"  Tafel mit Ideenbaum	UG SV

	S. sollen ein Verfahren finden, das o.a. Mangel behebt, Impuls dazu: "Betrachtet das jeweilige Textzeichen hinter dem Muster." Ggf. kann den Gruppen, die nicht weiterkommen, zusätzlich der Hinweis gegeben werden, aus diesem jeweiligen Zeichen und dem Muster Informationen zu gewinnen, um wie viele Positionen das Muster eigentlich weitergeschoben werden kann.	Muster "einige" und Text als Tischmodelle	PA
4) Sicherung	Ein S. simuliert das gefundene Verfahren an der Tafel und begründet, warum man jeweils um die angegebene Anzahl von Positionen weiter-schieben kann. L. nennt den Namen des Verfahrens und bittet um Erläuterung der Bezeichnung "vorausschauend" für dieses Verfahren. Die dahinterliegende Strategie wird als rechter Sohn von BF in den Ideenbaum eingetragen.	Tafelmodell  Tafel	SV  UG

Hausaufgabe	Je nach Zeit und Schülerwunsch kann das Verfahren für ein zweites Muster bei gleichem Text noch mal simuliert werden, wieder mit Bestimmung der jeweiligen Shift-Werte durch bloßes Hinsehen. Alternativ kann die Frage, um wieviel das Verfahren hier schneller ist, durch die Bestimmung der jeweiligen Vergleichsanzahlen angerissen werden. Dabei kann die Bestimmung bei BF durch scharfes Hinsehen erfolgen. Für QS müßte noch mal ein "Schnelldurchlauf" erfolgen.	Tafelmodell	UG / SV
	L. gibt Ausblick auf die nächste Stunde. L. teilt Kopien der Folie mit der umgangssprachlicher Beschreibung von BF aus der vorherigen Stunde aus. S. sollen daraus eine umgangssprachliche Formulierung von QS gewinnen.		LV

### Didaktisch-methodische Diskussion:

Die Anknüpfung an die letzte Stunde zu Beginn soll allen Schülern noch mal das Vorhaben und das bis dahin Erreichte in Erinnerung rufen.

Die generelle Konzeption dieser Stunde ergibt sich aus dem Verlauf der letzten Stunde und dem Begründungszusammenhang (2.6): Der Einbau von BF in das Rahmenprogramm soll möglichst als Bestätigung, daß das entwickelte Verfahren auch funktioniert, **vor** den Überlegungen zur Weiterentwicklung geschehen. Daher soll der erste Teil der Stunde zunächst genutzt werden, um die Hausaufgabe eines Schülers, die auf Folie geschrieben ist, gemeinsam zu besprechen und gegebenenfalls zu korrigieren, falls algorithmische Fehler darin sind. Diese sind höchstens im Bereich der Indexberechnung bei den nullterminierenden Strings zu erwarten, so daß ich kleine leere Haftschilder zu Verfügung stelle, auf die Variablennamen der Schülerlösung geschrieben und die unter die jeweils zu vergleichenden Text- und Musterpositionen geheftet werden können, um gegebenenfalls strittige Punkte am Modell zu klären.

Bei der Implementation können die Schüler dann ihre Version von BF gegebenenfalls analog zur Besprechung korrigieren und testen. Daher habe ich auch das Austesten nicht zusätzlich als Hausaufgabe zu dieser Stunde gegeben, ohne vorher überhaupt Aspekte der Implementation gemeinsam besprochen zu haben.

Die nochmalige Darlegung der Idee von SB soll diese einerseits wieder in Erinnerung rufen und ihren Effekt als eine mögliche Verbesserungsrichtung herausstellen, andererseits soll der Eintrag in den Ideenbaum den Blick für die spätere zweite Verbesserungsrichtung weiten.

Falls kein weiterer Nachteil von BF genannt wird, könnte ich einen Schüler noch mal eine Simulation an der Tafel durchführen lassen und mit der Frage, ob das nicht langweilig sei, auf die Monotonie des Weiterschiebens in BF hindeuten. Meiner Erwartung nach dürfte dies aber eigentlich nicht nötig sein.

Die Methode des gelenkten Entdeckens ist für das Finden der Grundidee von QS angemessen, da dieser Ideenbereich einerseits sehr weit ist, d.h. Verbesserungen im Bereich schnelleren Weiterschlebens sind zunächst schwierig selbst zu erfinden; andererseits ist dieser Ideenbereich recht unüberschaubar, d.h. es gibt viele Ideen, die schwierig, langwierig oder gar nicht effizient umzusetzen sind, ohne daß dies gleich offensichtlich wäre (s. etwa KMP, S.9). Nach der Erfahrung der letzten Stunde reicht der angegebene Impuls aus, er ist so offen wie möglich formuliert und kann bei Bedarf für einzelne Gruppen präzisiert werden.

Die später zu thematisierende Shift-Tabelle bleibt zunächst außen vor, die Schüler ermitteln jeweils durch Hinsehen den zum Textzeichen gehörigen Shift-Wert. Die Überlegungen zur genauen programmiersprachlichen Umsetzung mit Hilfe des `Array` und der Vorverarbeitung des Musters schließen sich in der nächsten Stunde folgerichtig an. Ich hatte zunächst überlegt, die so abgelesenen Shift-Werte parallel zur Simulation von QS in Form einer Tabelle auf die zweite Tafel notieren zu lassen, um so die Implementation schon vorzuformen. Es könnte dann aber der falsche Eindruck entstehen, daß die Shift-Werte parallel zum Suchen berechnet werden mit jedesmaliger Abfrage, ob der aktuell benötigte Wert schon mal berechnet wurde. Außerdem befinden sich die Schüler in dieser Phase auf der Ebene der enaktiven Darstellung, während die Tabelle schon in Richtung der ikonischen bzw. sogar symbolischen Darstellungsebene geht. Ferner würde das parallele Notieren die umgangssprachliche Beschreibung des Verfahrens erheblich erschweren.

Im Muster "einige" kommen zwei Buchstaben doppelt vor, so daß den Schülern klar werden muß, daß das jeweils letzte Auftauchen eines Buchstabens für dies Verfahren entscheidend ist.

Bei der Simulation sorgt die jeweilige Begründung der angewandten Sprungweite für eine sichere Verankerung des Verfahrens. Die Bezeichnung "vorausschauend" für QS habe ich einem c't-Artikel über Mustersuche entnommen,<sup>27</sup> was ich den Schülern auch mitteilen werde. Da einige Schüler diese Zeitschrift lesen, sollte das für sie motivierend sein, weil sonst eher selten Gegenstände des Lernbereichs Algorithmik Bezug zu aktuellen Publikationen haben. Die Erläuterung soll noch mal die Idee, die hinter QS steckt, herausstellen und zur Notation dieser Idee im Baum überleiten.

Die Hausaufgabe ist integraler Bestandteil der Stunde. In der Stunde wird lediglich die enaktive Darstellung von QS angestrebt, die schriftliche Sicherung in Form der umgangssprachlichen Beschreibung erfolgt zur nächsten Stunde. Diese findet schon am nächsten Tag statt, so daß das Fehlen einer schriftlichen Sicherung in dieser Stunde unproblematisch ist.

Am Stundenende kann durch das Gegenüberstellen der Vergleichsanzahlen eine schöne Zwischenbilanz gezogen werden, da ich Beispieltext und Bei-

---

<sup>27</sup> s. Gronek, S.278

spielmuster entsprechend gewählt habe (BF: 49 Vergleiche, QS: 16 Vergleiche).

## **4.2 Durchführung**

Die Zusammenfassung von Jan ergänzt Dirk um die Anwendungsfälle und die Abgrenzung der Mustersuche. Eike legt seine Folie aus der Hausaufgabe auf und erläutert sehr schön seine PASCAL-Funktion, indem er für einzelne Blöcke, teilweise unter Zuhilfenahme des Tafelmodells, klarmacht, was passiert, ohne zu sehr ins Detail zu gehen. Die Funktion ist leider etwas länglich, da er unterschiedliche Rückgabe-Codes, etwa für den Fall, daß das Muster länger als der Text ist, vorgesehen hat. Die Funktion ist gut lesbar und im wesentlichen korrekt, so daß es keine Fragen oder Anmerkungen der Mitschüler gibt und die Variablenschilder nicht benötigt werden.

Phase 2 benötigt mehr Zeit als eigentlich geplant, da beim Testen der in der Hausaufgabe entwickelten Versionen einige Fehler gefunden werden und deren Beseitigung viel länger dauert als vom Lehrer eigentlich vermutet. Er entschließt sich, diese Phase auszudehnen. Nachdem insgesamt 25 Minuten der Stunde vorbei sind, sind 3 Schüler komplett mit der Programmierung fertig, bei zweien ist lediglich die Abbruchbedingung noch nicht ganz korrekt, und die beiden anderen sind auf dem richtigen Weg.

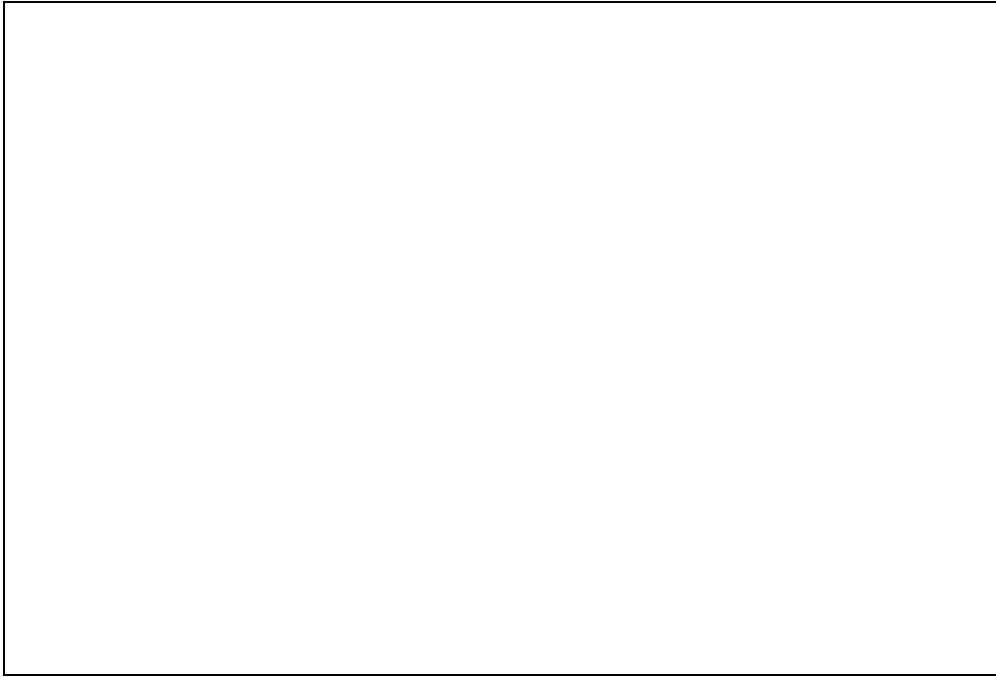
Der Lehrer bricht die Programmierphase hier ab und erteilt die restliche Implementation als ersten Teil der Hausaufgabe.

Die Phasen 3 und 4 verlaufen besser als erwartet: Dirk erläutert, wie erhofft, sehr pointiert die Idee von SB am Modell und nennt gleich den erzielten Effekt dazu; dies wird in den Baum eingetragen, dessen Wurzel schon an der Tafel steht. Der Nachteil, daß BF immer nur um eine Position schiebt, wird von Eike ohne weiteres erkannt, aber dann schauen die Schüler etwas ungläubig und denken nicht weiter, da sie wohl keine Vorstellung davon haben, wie man diesen Nachteil beheben kann, d.h. ob und gegebenenfalls wie man Sprünge machen kann. Als der Lehrer den beschriebenen Impuls gibt, schaut Stefan kurz in die Luft und macht dann seiner Erkenntnis mit einem laut in den Raum gerufenen "Aah" Luft...

Keine der Gruppen benötigt weitere Hilfestellung, alle kommen auf die fragliche Idee, einige schieben manchmal im Tischmodell zu weit, korrigieren sich bei kritischer Nachfrage aber selbst.

Zwei Gruppen, die schon fertig sind, rufen sich gegenseitig ihre Vergleichszahlen zu, die bis auf leichte Zählfehler übereinstimmen.

Schon nach kurzer Zeit führt Björn, ansonsten ein etwas stillerer Schüler, das neue Verfahren sehr gut und pointiert vor und liefert dabei sogar ungefragt jeweils die Begründung, warum er um die angegebene Anzahl von Positionen weiterschieben darf.



Björn simuliert das neue Verfahren QS.

Der Ausdruck "vorausschauend" wird von Stefan in Richtung "der Algorithmus schaut nach vorn" (d.h. nach rechts) gedeutet. Markus K. ergänzt, daß der Algorithmus schon den Ausgang einiger Vergleiche voraussehe, was im Baum als rechter Sohn des BF-Knotens notiert wird.

Dirk setzt unaufgefordert die Vergleichsanzahl von QS gegen die von BF, die er durch scharfes Hinsehen und Abzählen mit Blick auf die Tafel bestimmt, und Markus R. reit schon eine Kopplung der beiden Verfahren zur Optimierung an, was entsprechend gewrdigt und als Thema einer spteren Stunde deklariert wird.

Da die Stunde fast zu Ende ist und das neue Verfahren nach der guten Erluterung allen klar zu sein scheint, wird auf eine zweite Durchfhrung verzichtet, der Ausblick kann ebenfalls kurz ausfallen. Whrend der letzten paar Minuten kopiert der Lehrer den Schlern ihre aktuellen Versionen von BF fr die Hausaufgabe auf Disketten.

Nach Ende der Stunde kommt Markus R. zum Lehrer und beschwert sich, da die Vergleichsanzahl bei QS ja "Betrug" sei, da man ja jedesmal fr die Ermittlung des Shift-Wertes das Muster durchgehen msse. Auf die Frage des Lehrers, ob man das wirklich jedesmal msse, umreit der dabeistehende Jan die Realisierung dieser Abfragen mittels einer einmaligen Vorverarbeitung des Musters und einer Mengen-Variable, in der alle verschiedenen Buchstaben des Musters inklusive der zugehrigen Shift-Werte gespeichert werden, und Markus R. stimmt ihm zu. Der Lehrer fragt, ob man das nicht auch etwas simpler abspeichern knne, woraufhin die beiden Schler diskutierend in die Pause gehen. Durch diese Hinweise werden die beiden Schler zum Nachdenken angeregt, ohne schon Inhalte der nchsten Stunde fr sie vorwegzunehmen.

### 4.3 Reflexion

Die kleinen Schwächen bei der Programmierung haben gezeigt, daß die Schüler doch nicht so sicher darin sind, wie zu Anfang angenommen. Dies zeigte sich in einigen kleinen Syntaxfehlern auf der Hausaufgabenfolie, vor allem aber darin, daß die Programmierphase sehr viel mehr Zeit in Anspruch genommen hat als eigentlich geplant. Meine Entscheidung, diese Phase nach einer gewissen Zeit abubrechen, obwohl noch nicht alle Schüler fertig waren, erscheint mir jedoch auch im Nachhinein gerechtfertigt, da die Weiterführung der Programmierung als Hausaufgabe hier eine sinnvolle Fortführung der unterrichtlichen Tätigkeit ist. Das erste Lernziel kann somit erst in der nächsten Stunde überprüft werden. Der Zeitpunkt des Abbruchs war angemessen gewählt, einige Schüler waren mit dem Programmieren fertig, alle anderen auf dem richtigen Weg, und zusätzlich konnten alle Intentionen bezüglich QS realisiert werden.

Die Entwicklung von QS verlief sehr produktiv und zügig. Der Impuls war angemessen, was sich darin zeigt, daß die Schüler zunächst keine Vorstellung davon hatten, wie man größere Sprünge beim Verschieben machen kann, aber mit dem Tip doch alle zum Ziel gekommen sind. Stefans Heureka-Erlebnis stellt dies besonders deutlich dar.

Erfreulich war auch die sehr gelungene Präsentation von Björn, die er gänzlich selbständig und ohne weiteres Nachfragen meinerseits gestaltete und die zum guten Verständnis des Verfahrens bei den Mitschülern nach deren Bekunden beitrug.

Interessant war auch, daß Markus R. schon in dieser Phase die beiden Ideen koppeln wollte, was die gute Handhabbarkeit der Ideen in diesem Themengebiet zeigt, die Offenheit der Grobplanung im nachhinein rechtfertigt und auch als positiver Effekt des Ideenbaums gesehen werden kann, da man mit dessen Hilfe gut über die Ideen und deren Verknüpfung sprechen kann.

Der Abbruch der Programmierarbeit war so auch für die Schüler angemessen und vernünftig. Die Planung der Programmierarbeit für den späteren Teil der Stunde hätte einen glatteren Übergang zur Hausaufgabe ergeben, hätte den Nachteil gehabt, daß die erste Verbesserung (QS) schon erarbeitet würde, ohne daß die Implementation von BF klar wäre (vgl. Didaktisch-methodische Diskussion); daher ist die hier gewählte Reihenfolge günstiger gewesen.

Als Konsequenz ergibt sich insgesamt, daß bei der Planung der Implementation von QS mindestens eine gemeinsame Besprechung notwendig ist, was ich auch aus dem Gespräch mit Markus R. und Jan nach der Stunde ableite: Die beiden hatten die nötige Vorverarbeitung des Musters schon richtig erkannt, hatten aber eine Mengen-Implementation vor Augen, was wahrscheinlich durch die Sprechweise der Schüler bei der Entwicklung und Simulation von QS ("Wenn das Zeichen im Muster nicht vorkommt, kann man ganz dran vorbeischieben, sonst muß man nachsehen, wo im Muster das Zeichen steht") ergeben hat. Aber eine Korrektur der Sprechweise wäre in diesem Stadium der

Entwicklung unangebracht - weil für die Schüler nicht einsichtig - gewesen, so daß ich die daraus resultierenden zu erwartenden Schwierigkeiten in die Planung der nächsten Stunde einbeziehen werde.

## 5 Die dritte Stunde

### 5.1 Planung

#### Thema der Stunde:

Planung und Realisierung der programmiersprachlichen Umsetzung des QS-Verfahrens

#### Intentionen:

Die Schüler setzen das durch Simulation und umgangssprachliche Formulierung bekannte Verfahren QS in PASCAL um. Sie erkennen das `Array` als Möglichkeit, die aus dem Muster gewonnen Informationen zu speichern und effizient abzurufen.

Sie planen dazu die nötige Datenstruktur, erkennen und lösen die Probleme bei der Vorverarbeitung und benennen die in der eigentlichen Suchprozedur nötigen Veränderungen und nehmen sie vor.

Sie präsentieren die Ergebnisse ihrer Partnerarbeit, stellen sie zur Diskussion und wählen gemeinsam die günstigste Lösung aus. Falls genug Zeit bleibt, beginnen sie mit der Programmierung.

#### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Anknüpfung an vorherige Stunde	Ein S. stellt seine umgangssprachliche Formulierung des Verfahrens QS vor. Ein S. erläutert den Zusammenhang von QS und BF. Ein S. zeigt an der Folie vom letzten Mal die Stelle in der Prozedur, die verändert werden muß, wobei noch nicht klar ist, wie genau die Veränderung lauten wird.	HA-AB  Folie der letzten Stunde	SV

2) Erarbeitung I / Sicherung I	<p>Für das Muster "einige" der letzten Stunde, das an der Tafel hängt, erläutern S. einige Shift-Werte, diese werden an der Tafel in der Form</p> <table><tr><td>,</td><td>e</td><td>n</td><td>g</td><td>i</td><td>w</td><td>z</td></tr><tr><td>7</td><td>1</td><td>4</td><td>3</td><td>2</td><td>7</td><td>7</td></tr></table> <p>notiert und Speicherungsmöglichkeiten werden besprochen.</p> <p>Falls nötig, zieht L. zieht noch als nonverbalen Impuls den waagerechten und die senkrechte Striche:</p> <table><tr><td>,</td><td>e</td><td>n</td><td>g</td><td>i</td><td>w</td><td>z</td></tr><tr><td>7</td><td>1</td><td>4</td><td>3</td><td>2</td><td>7</td><td>7</td></tr></table> <p>Daraufhin sollte von S. das Stichwort <code>Array</code> fallen.</p>	,	e	n	g	i	w	z	7	1	4	3	2	7	7	,	e	n	g	i	w	z	7	1	4	3	2	7	7	Tafelmodell Tafel	UG
,	e	n	g	i	w	z																									
7	1	4	3	2	7	7																									
,	e	n	g	i	w	z																									
7	1	4	3	2	7	7																									
3) Erarbeitung II	<p>S. bearbeiten das AB als Vorbereitung für die Programmierung, es schreiben beide Partner jeweils mit, um ggf. eine Sicherung für die HA zu haben.</p> <p>Währenddessen spielt L. die in der HA vervollständigten BF-Prozeduren der S. auf das Netz in deren persönliche Verzeichnisse.</p>	AB (s. S.70)	PA																												
4) Sicherung II	Zusammentragen und Diskussion der Ergebnisse	AB	UG																												
5) Implementati- on	Falls angemessen viel Zeit bleibt, beginnen S. gemäß ihrer Überlegungen mit der Implementation.	AB Computer	EA																												
Hausaufgabe	S. vervollständigen die Implementation mit Hilfe der Ergebnisse auf dem AB.																														

### Didaktisch-methodische Diskussion:

Durch den in der Hausaufgabenbesprechung hergestellten Zusammenhang zwischen BF und QS sollte den Schülern klar sein, daß sie für die eigentliche Suchprozedur lediglich die BF-Prozedur kopieren und in Bezug auf das jeweilige Weiterrücken des Musters (in BF: `inc(i)`, in QS: `inc(i,...)`) abändern müssen. Daher setze ich hier auch keine Folie für die Präsentation der umgangssprachlichen Beschreibung ein, da diese sich eben nur geringfügig von der Folie aus der ersten Stunde unterscheiden würde und dieser kleine Unterschied für diese Schüler auch durch bloßes Vor- bzw. Mitlesen in den eigenen Aufzeichnungen faßbar ist.

Es ist jedoch nicht klar, was genau nun bei den Auslassungszeichen bei `inc(i,...)` stehen muß. Diese Unklarheit motiviert die weitere Untersuchung und leitet zur nächsten Phase über, in der noch mal einige Shift-Werte bestimmt werden und wo die prinzipielle Speicherung im `Array` besprochen wird. Dies Unterrichtsgespräch zur Klärung der Implementationsfragen ist not-

wendig, was einerseits die Schwierigkeiten bei der Programmierung zu Anfang der letzten Stunde, andererseits die im Gespräch nach Ende der letzten Stunde von Markus R. und Jan geäußerten Überlegungen gezeigt haben.

Die Realisierung der Speicherung mittels eines Mengen-Typs ist zwar möglich, jedoch langwierig, und eventuell ist QS dann wegen des komplizierten Zugriffs gar nicht wesentlich schneller als BF; dies sollte für die erste Verbesserung ja nicht der Fall sein.<sup>28</sup> Auch im Hinblick auf das Reihenziel, daß die Schüler einen Überblick über das Themengebiet Mustersuche erhalten sollen, möchte ich die Array-Implementation allen Schülern nahebringen, so daß das Array die Datenstruktur der Wahl in Phase 2 ist. Im Gespräch werden Index- und Wertetyp nicht genau erörtert, so daß den Schülern hier noch genug Spielraum für eigene Überlegungen in Phase 3 bleibt. Alternativ zur gewählten Darstellung an der Tafel hatte ich folgende etwas stärker an die bisherigen Vorstellungen angelehnte Notation in Betracht gezogen:

e	i	n	g	sonst
1	3	4	2	7

Diese Darstellung schien mir aber wieder zu sehr auf die Mengen-Vorstellung hinauszulaufen, so daß ich mich dagegen entschieden habe.

In Phase 3 sollen die genauen Implementationsdetails, d.h. genauer Datentyp, Vorverarbeitung und Veränderung der Suchfunktion, von den Schülern erarbeitet werden. Dazu erschien mir die Partnerarbeit als Sozialform günstig. Die Alternative, die ganze Besprechung der Programmierung von QS im Plenum zu vorzunehmen, habe ich ausgeschlossen, da das Unterrichtsgespräch einfach zu lang werden würde; die andere Alternative, die Entwicklung der programmiersprachlichen Umsetzung ganz ohne Plenumsphase zu planen, lehne ich aus den oben genannten Gründen ab. Der hier getroffene Kompromiß scheint angemessen, er garantiert einen Phasenwechsel, und in der Partnerarbeit können auch schneller Diskussionen über mögliche Problemlösungen zustande kommen, weil deren Zielrichtung in Phase 2 geklärt wurde. Die Aufträge auf dem Arbeitsblatt (s. S.70) sind dazu offen, aber zielgerichtet formuliert.

Mit der Hausaufgabe aus dieser Stunde ist dann die Entwicklung und Implementation von QS zunächst abgeschlossen.

## 5.2 Durchführung

Dirk ist einige Minuten vor Stundenbeginn im Informatikraum und beschwert sich beim Lehrer, daß die Vergleichsanzahl bei QS in der letzten Stunde ja nicht ehrlich bestimmt worden sei mit dem selben Hinweis wie Markus und Jan nach Ende der letzten Stunde (s. S.33), was ihm der Lehrer als Thema dieser Stunde bekanntgibt.

---

<sup>28</sup> s. Auswahl der Verfahren, S.14

Zu Anfang der Stunde zeigen sich die Schüler wie erwartet aufgrund des Stundenzzeitpunkts (Freitag 6.Stunde) und der Klausurphase etwas träge, und es stellt sich heraus, daß einige Schüler die Fertigstellung der Implementation von BF in den Hausaufgaben nicht und daß die umgangssprachliche Formulierung nur ein Schüler gemacht hat. Nach einer entsprechenden Bemerkung vom Lehrer dazu liest Dirk seine richtigen Hausaufgaben vor, die jedoch noch die Formulierung "wenn das Zeichen nicht im Muster ist ..." aufweist.

Jan R. zeigt zunächst eine falsche Stelle an der Folie, die zu ändern ist, Eike hilft und merkt an, daß man "da um mehr als Eins erhöhen muß, aber ... Naja... Wie denn jetzt?".

Es ergibt sich so insgesamt ein schöner Einstieg, da genau dies Ansatzpunkt der in dieser Stunde zu machenden Überlegungen ist.

Bei der Notation der Shift-Werte des an die Tafel gehefteten Musters schlägt Jan noch mal den Datentyp Menge vor. Bei dem Hinweis, daß der Datentyp `set` in PASCAL sehr ineffizient implementiert sei, und bei der Frage nach möglichen Alternativen hat keiner der Schüler eine Idee. Erst bei dem graphischen Impuls des Tabellenzeichnens nennt Dirk das Stichwort `Array`.

Der Lehrer teilt die Arbeitsblätter (s. S.70) aus, und die Partnerarbeit in Phase 3 verläuft dann lebhafter.

#### Bearbeitung von Aufgabe 1:

Stefan schlägt in seiner Gruppe einen selbstdefinierten Indextyp für das `Array` vor, der alle (verschiedenen) Zeichen des Musters aufnehmen kann, was Markus K. mit dem Hinweis, daß man dann einen variablen Typ haben müsse, und der Frage, ob es nicht einfacher gehe, ablehnt. Auf den strategischen Tip vom Lehrer hin, es möglichst simpel zu machen, richten einige Gruppen als Indextyp den Ausschnittstyp `0..255` ein, andere fragen ungläubig nach, ob denn `Char` als Indextyp zulässig sei, was vom Lehrer bestätigt wird. Auch der Ausschnittstyp `chr(0)..chr(255)`, der mit dem Typ `Char` übereinstimmt, wird verwendet.

#### Bearbeitung von Aufgabe 2:

Bei den Überlegungen zur Wertbelegung des `Array` taucht eine Version auf, die für jedes mögliche Textzeichen das ganze Muster durchgeht und das `Array` mit dem entsprechenden Wert belegt, auf; ebenso gibt es eine Version mit einer `For`-Schleife, die das Muster durchgeht, und einer `IF`-Abfrage, ob das fragliche Zeichen schon mal vorkam. Zwei Gruppen entwickeln schließlich die Version mit einer `For`-Schleife, in der schon belegte Werte, die von einem weiter links stehenden Auftreten des jeweiligen Zeichens stammen, einfach überschrieben werden.

#### Bearbeitung von Aufgabe 3:

Einigen der Gruppen bereitet der doppelte `Array`-Zugriff innerhalb des Ausdrucks `Schiebetabelle[Text[T+Musterlaenge]]` Schwierigkeiten. Für die Gruppe, die das `Array` mit `0..255` deklariert hat, kommt ein Funktionsaufruf von `ord` innerhalb des Ausdrucks hinzu.

Die Gruppen besprechen auch die Frage, ob man nicht `Text[T+Musterlaenge+1]` nehmen müsse, um das Textzeichen **hinter** dem Musterende zu bekommen. Die Gruppen nehmen Tischmodelle aus den vorherigen Stunden zum Abzählen zur Hilfe.

Die Besprechung der Aufgaben in Phase 4 verläuft sehr produktiv, es werden alle jeweils erarbeiteten Ergebnisse genannt und deren Vor- und Nachteile besprochen. Bei der Deklaration des `Array` wird `0..255` als Indextyp vorgeschlagen, um beim Initialisieren die Laufvariable der `For`-Schleife direkt zum Indizieren zu benutzen. Daraufhin schreibt Markus R. eine `For`-Schleife mit Laufvariable vom Typ `char` und Startwert `chr(0)` und Endwert `chr(255)` an die Tafel, was allgemein als beste Lösung anerkannt wird (s. auch Listing des erstellten Programms, S.82). Damit sind die ersten beiden Aufgaben des Arbeitsblattes erledigt und Markus R. leitet unaufgefordert und ohne weiteres zur Besprechung der dritten Aufgabe über. Er benutzt beim Programmieren meistens sehr kryptische Variablenbezeichner und schreibt als seinen Vorschlag zur Veränderung des BF-Algorithmus `inc(tIndex, ST[t[tIndex+Musterlaenge+1]])` an die Tafel. Zunächst wird er von seinen Mitschülern aufgefordert, die Variablennamen zu erläutern, dann wird anhand des zur Sicherheit an der hinteren Tafel befestigten Modells der letzten Stunde die Frage nach dem "+1" geklärt und dies weggewischt. Als graphische Unterstützung unterklammert der Lehrer den Ausdruck `t[tIndex+Musterlaenge]` und schreibt "Erstes Textzeichen hinter dem Muster" darunter.

Ein Schüler faßt noch mal alle nötigen Änderungen an BF zusammen. Da nur noch 2 Minuten Zeit sind, erteilt der Lehrer als Hausaufgabe für diejenigen, die die Hausaufgabe zur Stunde nicht gemacht hatten, die Fertigstellung von BF und für alle die Programmierung von QS gemäß der Besprechung in Phase 4.

### 5.3 Reflexion

Insgesamt bin ich mit der Stunde nicht voll zufrieden. Es konnten alle Intentionen realisiert werden, jedoch hatte ich eigentlich gedacht, daß zumindest vernünftige Anfänge des Programmierens noch in der Stunde gemacht würden.

Insgesamt war die Stunde nicht durch so viel Kreativität von Seiten der Schüler gekennzeichnet wie etwa die erste Stunde der Reihe. Das kann einerseits an dem Thema dieser Stunde liegen, andererseits an dem etwas eng geführten Unterrichtsgespräch zu Anfang. Aufgrund der Erfahrungen nach der letzten Stunde und dem Gespräch mit Dirk vor dieser Stunde war dies aber nötig. In den nächsten Stunde wird es, auch thematisch, möglich sein, mehr Raum für eigenverantwortlichere und eigenständigere Schülerarbeit einzuräumen.

Die Stunde hat auch etwas unter den Rahmenbedingungen der Klausurphase und der Tatsache, daß am selben Tag eine Klausur geschrieben wurde,

gelitten. Da jedoch 2 Unterrichtsstunden Zeit zwischen der Klausur und der Informatikstunde lagen, war nicht damit zu rechnen, daß die Schüler so erschöpft in den Unterricht kommen. Auch daß ein Teil keine Hausaufgaben hatte, war nicht zu erwarten, da sonst die Hausaufgabenmoral recht gut ist. Mit dem Einstieg in die Partnerarbeitsphase verlief der Unterricht dann sehr viel lebhafter, was auch an den Diskussionen innerhalb der Zweiergruppen zu erkennen war.

Offensichtlich lenkte die Tatsache, daß der Algorithmus für verschiedene Muster funktionieren soll, die Aufmerksamkeit der Schüler auf einen Datentyp, der verschieden viele Angaben aufnehmen kann, da Muster auch verschieden viele unterschiedliche Buchstaben haben können. Insofern war die Menge eigentlich ein nicht unpassender Gedanke. Die Idee, einen Datentyp mit einer festen Anzahl darin abzulegender Informationen zu benutzen und beispielsweise die Musterlänge nicht in der Anzahl der Informationen, sondern in den Einträgen selbst abzuspeichern, ist wohl zu mittelbar. Die bis dahin verwendete Sprechweise beim Weiterschieben des Musters tat ihr übriges.

Der `Array`-Impuls in Phase 2 war also richtig und angemessen, da die Schüler sonst die Mengen-Implementation versucht hätten und wohl nicht auf die im Endeffekt viel leichtere und effizientere Implementation mit dem `Array` gekommen wären. Man hätte hier die Implementation mittels Mengen-Typ noch genauer besprechen können, um noch klarer zu machen, warum das `Array` hier günstiger ist.<sup>29</sup> Dies wäre allerdings über den zeitlichen Rahmen der Stunde erheblich hinausgegangen, so daß ich diese Möglichkeit nicht wahrgenommen habe, was das Erreichen des zweiten Lernziels zumindest als fraglich erscheinen läßt.

Mit den Schwierigkeiten beim Bestimmen des Index' des betreffenden Textzeichens hatte ich so nicht gerechnet, die Schwierigkeiten ergaben sich aus der Verwendung der nullterminierenden Strings im Rahmenprogramm. Hilfreich war hier, daß das Modell aus der Vorstunde an der hinteren Tafel benutzt werden konnte.

Die Diskussionen beim Zusammenführen der Ergebnisse in Phase 4 waren gehaltvoll und die Entscheidungen für oder gegen eine Lösung waren von der Sache her fundiert begründet und gemeinsam gefällt, so daß die Phasen 3 und 4 als sehr gelungen bezeichnet werden können.

Ob die programmiersprachliche Umsetzung allen Schülern gelungen ist, kann erst zu Anfang der nächsten Stunde beurteilt werden. Ich gehe allerdings davon aus, da die vorzunehmenden Änderungen in dieser Stunde genau besprochen wurden und von allen verstanden zu sein schienen.

---

<sup>29</sup> Sowohl die Deklaration, als auch die Benutzung sind so erheblich einfacher als bei der Menge. Diese müßte einen `Record`-Typ als Elementtyp haben, wobei eine Komponente das Musterzeichen, eine andere Komponente der zugehörige Shift-Wert wäre. Zusätzlich bräuchte man zur Abfrage innerhalb des Suchverfahrens eine Menge, in der lediglich die im Muster vorkommenden Zeichen gespeichert sind.

## 6 Die vierte Stunde

### 6.1 Planung

#### Thema der Stunde:

Weiterentwicklungen von QS:

Koppeln zweier Ideen ("Vorausschau" von QS und "Seltenster-Buchstabe-zuerst" von SB zu QS-SB) und Weiterverfolgung einer Idee ("Vorausschau" von QS zu QS-D)

#### Intentionen:

Die Schüler sollen die Kopplung und die Weiterverfolgung von Ideen am Beispiel der Verbesserungen von QS verwirklichen. Dazu sollen sie die Verbesserungsrichtung mündlich umschreiben, deren Zusammenhang graphisch veranschaulichen und die zugehörigen Verfahren entwickeln und programmiersprachlich umsetzen.

Sie sollen in Gruppenarbeit selbständig ein größeres Arbeitsvorhaben realisieren, die erarbeiteten (Teil-)Lösung ihren Mitschülern vorstellen und vom Stand ihrer Arbeiten angemessen berichten.

#### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Anknüpfung an vorherige Stunde	L. erzählt S. von Dirks Bemerkung vor Anfang der letzten Stunde über die Vergleichsanzahl von QS (s. S.37) und bittet ihn um eine erneute Kommentierung.		LV / SV
2) Zielfestlegung a)	Der Strategiebaum der vorletzten Stunde wird von der hinteren Tafel hervorgeholt. Ein S. macht den Standort klar, und die mögliche Kopplung der beiden bisher erarbeiteten Ideen wird als linker, farbliche entsprechend doppelt markierter Sohn des Knoten QS notiert.	Tafel Strategiebaum	UG
b)	L. gibt stummen Impuls, indem er eine Kante vom QS-Knoten zum noch nicht vorhandenen rechten Sohn QS-D zeichnet. Die Möglichkeit der weiteren Ausnutzung der Idee "Vorausschau" wird von S. formuliert und entsprechend in den Ideenbaum eingetragen.		

3) Parallele Erarbeitung	<p>Der Kurs teilt sich in zwei Gruppen ein, jede soll einen der neu entstandenen Knoten realisieren. Als Hilfe erhält die Gruppe mit der Idee "Seltenster Buchstabe" ihr AB und später die entsprechende Unit mit der Häufigkeitstabelle über das Netz.</p> <p>Die Gruppe QS-D erhält ihr AB mit dem Impuls. Beide Gruppen können ihre Tischmodelle der letzten Stunde benutzen. Die Mitte der nächsten Stunde wird als Fertigstellungstermin vereinbart, und die Auflage, am Ende dieser Stunde der anderen Gruppe die verwendete Idee am Tafelmodell vorzuführen und vom Stand der Arbeiten zu berichten, wird vom L. gegeben.</p> <p>Während der GA-Phase spielt L. die HA der S. auf das Netz und präpariert die Tafel für die Simulation: Er heftet Text und Muster der vorletzten Stunde an und notiert die dort bestimmten Vergleichsanzahlen von BF und QS.</p>	<p>Gruppenspezifische AB (s. S.72f.) Computer Tischmodelle</p>	GA
4) Bericht und Simulation der einzelnen Gruppen	Die beiden Gruppen schildern die Idee, die hinter dem von ihnen erarbeiteten Algorithmus steckt, simulieren ihn an der Tafel, notieren die Vergleichsanzahl und berichten vom Stand ihrer Arbeiten.	Tafelmodell Tafel	SV

### Didaktisch-methodische Diskussion:

Die generelle Konzeption dieser Stunde ergibt sich aus der Grobplanung, aus den bisher von den Schülern erarbeiteten Ideen (SB, Kopplung von QS und SB) sowie aus der Tatsache, daß die Schüler in der übernächsten Stunde eine LK-Klausur schreiben. Daher möchte ich die Schüler in dieser Stunde möglichst selbständig arbeiten lassen (Phase 3), ohne sie jedoch hilflos im Regen stehen zu lassen. Die vierte Stunde der Reihe eignet sich gut für eine solche freiere Phase, da die Schüler hier einerseits das Handwerkszeug für die Entwicklung weiterer Verfahren haben (BF und QS als typische Verbesserung sind von allen verstanden), andererseits sind auch schon Verbesserungsrichtungen klar geworden.

Zum Stundenanfang möchte ich Dirk zu Wort kommen lassen mit nochmaliger Stellungnahme zu seiner Bemerkung über QS, da seine damalige Vorstellung von der programmiersprachlichen Realisierung ohne Vorverarbeitung bei vielen Kursteilnehmern vorherrschte und insofern noch mal klargestellt werden sollte. Die Kursatmosphäre ist so gut, daß keine hämischen Bemerkungen o.ä. zu erwarten sind.

Die Erarbeitung der Verbesserungen der zweiten Stufe soll durch den jeweiligen Eintrag in den Ideenbaum eingeläutet werden, damit die Schüler von

Anfang an eine richtige Einordnung ihrer Tätigkeiten in den Zusammenhang haben. Die Kopplung von QS und SB zu QS-SB wurde ja schon gegen Ende der zweiten Stunde von Markus R. angerissen, so daß auch die entsprechende Darstellung im Baum (Phase 2) problemlos sein sollte. Der stumme Impuls für den Knoten zu QS-D wird wahrscheinlich von den Schüler zunächst nur formal aufgenommen, d.h. sie werden den Eintrag des QS-Knotens "Sieh den Ausgang einiger Vergleiche voraus und mache Sprünge" steigern, ohne zu wissen, **wie** diese Idee zu realisieren ist. Dabei soll das anschließend auszuteilende Arbeitsblatt (s. S.73) helfen.

Die Gruppenarbeit, in der zwei Gruppen **verschiedene** Verfahren erarbeiten und realisieren, scheint mir hier angemessen:

Für die Entwicklung beider Verbesserungen durch alle Schüler reicht die Zeit nicht aus, und durch die Präsentationsphase am Schluß der Stunde sind beide Gruppen über die Arbeiten der jeweils anderen Gruppe gut informiert, so daß alle Schüler alle Lernziele erreichen können. Mit Hilfe der Sozialform Gruppenarbeit kann hier auch gut das Präsentieren von (Teil-)Lösungen einer Aufgabe und Arbeitsstandsberichten geübt werden.

Da vor allem Markus R. und Björn im Laufe der Reihe vehement ihre Idee SB vertreten haben, werden sie sicher gerne die Ideenkombination ausführen. Die Realisierung der Idee zu QS-D scheint zunächst unklar, so daß auch hier einige Schüler neugierig sein werden, wie sie umgesetzt werden kann. Die Gruppenbildung kann also einfach nach Neigung erfolgen.

Die gegebenen Hilfen für die Teilgruppen auf den Arbeitsblättern sind angemessen. Der Gruppe QS-SB brauche ich auf dem Arbeitsblatt lediglich die zu Verfügung gestellte Unit zu erläutern (s. S.72), da hier sonst keine weiteren Hilfen angebracht sind. Diese Unit enthält die Deklaration und die Belegung des Häufigkeits-Array. Wie die Häufigkeiten zustande gekommen sind, werde ich von meiner Seite aus zunächst nicht thematisieren, diese Thematik soll jedoch spätestens in der Reflexionsstunde am Schluß der Reihe diskutiert werden.<sup>30</sup>

Der Impuls für die Gruppe QS-D auf dem Arbeitsblatt ("Sieh dir zusätzlich das Textzeichen (..) direkt über dem Musterende an") kann meines Erachtens nicht offener formuliert sein (z.B. "Betrachte weitere Textzeichen"), da die Schüler sonst mit großer Wahrscheinlichkeit das zweite Textzeichen hinter dem Musterende als zusätzliche Informationsquelle betrachten würden. Dies bringt aber keinen weiteren Gewinn, was aber erst nach sehr langwierigen und zeitintensiven Beispielen und Überlegungen klar wird, da man hier sehr viele verschiedene Möglichkeiten findet, **in Spezialfällen** das zweite Textzeichen ins Spiel zu bringen, was im allgemeinen Fall aber keinen Gewinn bringt. Da auch wegen der bevorstehenden Osterferien und der Synchronisation mit der anderen Gruppe soviel Zeit nicht zur Verfügung steht, wird als Tip das zusätzlich zu

---

<sup>30</sup> Die Häufigkeitstabelle, die ich den Schülern zu Verfügung stelle, habe ich aus der Analyse des vorgesehenen Test-Textes gewonnen.

betrachtende Zeichen vorgegeben, jedoch auf dem Arbeitsblatt nichts darüber gesagt, **wie** es in den Algorithmus einzubeziehen ist.

Die Präsentation am Schluß der Stunde dient dazu, alle Kursteilnehmer und den Lehrer grob über den Verlauf beider Gruppenarbeiten zu informieren.

Eine noch freiere Konzeption der Stunde, etwa ohne die Skizzierung im Ideenbaum oder die Vorgabe einer Präsentation, scheint mir unangemessen, da die konkrete Formulierung der zu verfolgenden Ideen den Schülern wertvolle Orientierung innerhalb der Unterrichtsreihe geben kann.

Angesichts der Klausurphase verzichte ich in dieser Stunde auf Hausaufgaben, obwohl etwa die umgangssprachliche Formulierung des Verfahrens der jeweils anderen Gruppe eine mögliche Hausaufgabe wäre.

## **6.2 Durchführung**

Die Schüler unterhalten sich zu Stundenbeginn noch über ihre bevorstehende Mathematik-LK-Klausur. Der Bericht vom Gespräch mit Dirk bringt aber sehr schnell die gewünschte Aufmerksamkeit. Dirk nimmt nochmals Stellung, indem er kurz die Ergebnisse der letzten Stunde skizziert.

In Phase 2a beschreibt Björn sofort die Kopplung der beiden Ideen, und sein richtiger Vorschlag der Notation in einem zweifarbig umrandeten Knoten<sup>31</sup> löst aufgrund der - vom Lehrer allerdings unbeabsichtigten - Farbwahl (rot-grün) leichte Erheiterung bei den Schülern aus.

Wie vermutet erntet der Lehrer auf den stummen Impuls in Phase 2b zunächst nur Schweigen. Nach der Bitte um Kommentierung des Pfeils sagt Eike etwas zögerlich: "Naja, da müßte stehen: 'Sieh einige Vergleiche mehr voraus und mache größere Sprünge'." Dies wird an der Tafel als Knoten notiert, von Jan als "rot-rot" tituliert und entsprechend umrandet.

Bei der Gruppenbildung melden sich spontan Markus R., Björn und Jan für die Gruppe QS-SB; Eike, Stefan und Dirk wählen QS-D, Markus K. fragt kritisch nach, "wie denn wohl 'rot-rot' gehen soll". Der Lehrer verweist darauf, daß auf dem Arbeitsblatt ein Tip zu finden sei, und Markus K. entscheidet sich ebenfalls für QS-D.

Die Gruppen erhalten die angegebenen Materialien, geben ihre Disketten mit der Hausaufgabe ab, die der Lehrer auf das Netz bringt, und nehmen ihre Arbeit auf.

Die Gruppe QS-D überlegt zunächst anhand der Tischmodelle noch kurz an Versionen zum Einbeziehen des zweiten Textzeichens hinter dem Muster. Dabei werden wie erwartet einige Spezialfälle betrachtet. Dann wird jedoch das Arbeitsblatt zur Hilfe genommen und an dem Beispiel überlegt, wie das Textzeichen über dem Musterende einzubeziehen ist. Zunächst wird dabei fälschlicherweise einfach das Maximum aus beiden Shift-Werten genommen, aber bei der Implementation am Computer machen sich die Schüler unter Zuhilfenahme des Tischmodells der vorletzten Stunde klar, daß der um Eins

---

<sup>31</sup> s. Tafelbild zur Stunde, S.71

verminderte Shift-Wert zu betrachten ist. Die Gruppe ist kurz vor der vereinbarten Zusammenführung fertig und hat eine lauffähige Version vorzuweisen. Diese wird dem Lehrer vorgeführt und Markus K. schließt direkt die Frage an, wie man denn nun feststellen könne, ob das schneller sei, was vom Lehrer als Inhalt der nächsten Stunden deklariert wird.



Die Gruppe QS-D bei der Überprüfung ihrer Lösung mittels Tischmodell.

Die Gruppe QS-SB setzt sich in dieser Stunde noch gar nicht an den Computer, sondern plant zunächst ihre Vorhaben, was besonders für Markus R., der sonst meist gleich losprogrammiert, recht ungewöhnlich ist.



Die Gruppe QS-SB plant ihr Verfahren.

Jan schlägt vor, doch zunächst den seltensten Buchstaben in einer Vorverarbeitung zu bestimmen und diesen dann ganz normal im Text zu suchen. Markus R. weist ihn darauf hin, daß das doch wohl "nur grün" (gemeint ist der grüne Knoten im Ideenbaum) sei. Markus R. schlägt nun vor, den seltensten Buchstaben als Index für den Zugriff auf die Shift-Tabelle zu verwenden. Erst der strategische Hinweis von Björn ("Laßt uns doch erstmal das Einfachste machen und dann weitersehen!") führt weiter. In dieser Gruppe gibt es heftige und sehr kontroverse Diskussionen um die Implementation, wobei es aber stets um die Sache geht.<sup>32</sup> Als die Gruppe die Lösungsschritte grob besprochen hat, wird noch die Vor-Prozedur, die mit Hilfe der vorgegebenen Häufigkeitstabelle den seltensten Buchstaben im Muster bestimmt, auf Papier korrekt programmiert.

In Phase 4 führt Jan noch mal die Idee von SB kombiniert mit der von QS vor, notiert die entsprechende Vergleichsanzahl und berichtet kurz vom Stand der Entwicklung.

Stefan erläutert sehr schön die Idee von QS-D, simuliert sehr pointiert und erklärt die Implementation mittels  $\text{if}$ -Abfrage. Diese wird von ihm auch gleich kritisch bezüglich der Schnelligkeit beleuchtet, und Eike stellt fest, daß man diese  $\text{if}$ -Abfrage eventuell sogar bei den Vergleichen mitzählen müßte. Der Lehrer weist darauf hin, daß es hier noch um Muster-Text-Vergleiche gehe, aber diese Bedenken in der letzten Stunde der Reihe noch einmal vorgebracht werden sollten.

Als Stefan die Vergleichsanzahl notiert, klingt die Stunde mit dem augenzwinkernden Hinweis von Dirk aus, daß das Beispiel vom Lehrer ja wohl so gewählt

---

<sup>32</sup> Auf die Frage des Lehrers an Björn, ob sie sich streiten würden, antwortet dieser: „Ja, so geht’s am besten, so kommt am meisten raus!“

sei, daß die beiden Verbesserungen von QS auf dieselbe Vergleichszahl kommen.

Die Analyse der Hausaufgaben in den Schülerverzeichnissen ergibt später, daß alle die Hausaufgabe gemacht haben, bei allen die BF-Version richtig ist und nur bei wenigen in der QS-Version kleine Fehler bei der Vorverarbeitung vorliegen.

### **6.3 Reflexion**

Der breite Raum für Schüleraktivitäten in dieser Stunde hat sich bezahlt gemacht: Die Schüler waren trotz der bevorstehenden Klausur mit großem Engagement, Begeisterung und Konzentration bei der Sache, was sich auch durch die Sozialform Gruppenarbeit ergab. In dieser Stunde wurde besonders deutlich, wieviel Spaß ihnen die Sache macht.

Der Impuls für den QS-D-Knoten löste zunächst etwas Verunsicherung aus, die dann in Neugier umschlug, so daß dies Vorgehen neben der Orientierung der Schüler noch weitere positive Wirkung hatte. Die farbige Markierung der Knoten wurde von den Schülern sofort aufgenommen, die Gruppen titulierte sich gegenseitig als "rot-grün" und "rot-rot".

Im Nachhinein wäre zu überlegen, ob der Impuls für QS-D nicht doch freier hätte formuliert werden können, dann hätte diese Gruppe aber sicher **sehr** viel mehr Zeit benötigt und sich eventuell in Spezialfälle verbissen. Dies hätte man vielleicht durch die strategische Hilfe, eine möglichst allgemeine Methode zu finden, oder durch ein speziell für diese Gruppe gewähltes neues Beispiel auffangen können.

Insgesamt bin ich mit dieser Stunde sehr zufrieden, alle Ziele wurden erreicht, was ich dem teilweise direkt aus den Schülerantworten entstandenen Tafelanschrieb, meinen Beobachtungen während Phase 3 und der gelungenen Präsentationsphase 4 entnehme. Über die lange Vorlaufzeit bei der Gruppe QS-SB war ich etwas verwundert. Die Idee war eigentlich hier schon klar, und daher hatte ich damit gerechnet, daß diese Gruppe in dieser Stunde schon mit dem Programmieren am Computer anfängt. Die akribische Planung und Vorarbeit war aber nötig und sinnvoll, was an den Diskussionen innerhalb dieser Gruppe deutlich wurde. Demgegenüber ist die Gruppe QS-D wesentlich schneller als erwartet fertig geworden. Für die nächste Stunde ergibt sich daraus der Vorteil, daß nach der gemeinsamen Testplanung die Gruppe QS-D die nötigen Testvorbereitungen schon treffen kann, während die andere Gruppe ihr Vorhaben noch fertigstellen soll.

Für die Programmierarbeiten zur Testvorbereitung muß diejenige Version des Rahmenprogramms vorliegen, die später zum Testen verwendet wird. Dazu muß aus den Schülerversionen auf dem Netz eine Version ausgewählt werden. Das wäre auch eine interessante Aufgabe für die Schüler gewesen, man hätte unterschiedliche Gesichtspunkte zum Vergleich erarbeiten und anwenden können. Da dies nicht unmittelbar zum Thema der Reihe gehört, auch

nicht in den zeitlichen Rahmen der Reihe paßt und da ohnehin viele gute und fast gleichartige Versionen vorliegen, treffe ich diese Auswahl. Die Version von Markus R. ist kurz und gut verständlich, es wird noch QS-D aus Markus K.'s Version hineinkopiert und die so vervollständigte Fassung in allen Schüler-Verzeichnissen abgelegt.

## 7 Die fünfte Stunde

### 7.1 Planung

#### Thema der Stunde:

Gemeinsame Testkonstruktion, Durchführung der Testvorbereitung durch die Gruppe QS-D und Vervollständigung der Gruppenarbeit QS-SB

#### Intentionen:

Die Schüler sollen die Testbedingungen für den abschließenden Vergleichstest festlegen. Sie sollen dabei Vergleichskriterien auswählen, Bedingungen an Text und Muster formulieren, auf die Gegebenheiten (verfügbare Zeit, vorhandener Text) abstimmen, d.h. diese genauso übernehmen, anpassen oder nötigenfalls fallenlassen, und das bisherige Rahmenprogramm entsprechend erweitern.

Sie sollen dabei die Problematik der Text- und der Musterwahl für den Test erkennen und eine Lösung dafür angeben.

#### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Einstieg	L. gibt Überblick über die Stunde und erklärt die Auswahl von Markus R.'s Version, die ab jetzt weiterbearbeitet werden soll.		LV
2) Erarbeitung / Sicherung	Die für den Test zu beachtenden Punkte inklusive der daraus sich ergebenden Vorbereitungen werden erarbeitet und an der Tafel festgehalten.	Tafel	UG
3) Parallele GA: Testvorbereitung Erarbeitung	Die Gruppe QS-D führt die besprochenen Vorbereitungsarbeiten aus. Die Gruppe QS-SB vervollständigt ihre Arbeit aus der vorherigen Stunde.	Computer Tafel, Medien der letzten Stunde	GA  GA
4) Zusammenfassung Hausaufgabe	Ein S. faßt noch mal die Testmodalitäten zusammen und erläutert die HA für die Mitschüler. Jeder S. erhält zwei Musterlängen zugeteilt, für die er sich Muster gemäß den Absprachen ausdenkt. Zur nächsten Stunde sind Taschenrechner für die Durchschnittsbildung mitzubringen.	Tafel	SV

### Didaktisch-methodische Diskussion:

Die fünfte Stunde der Reihe besteht im wesentlichen aus zwei Teilen: In Phase 2 wird gemeinsam der in der nächsten Stunde durchzuführende Test geplant. In Phase 3 werden die beiden Gruppen getrennt arbeiten: Während die Gruppe QS-D in der letzten Stunde ihre Aufgabe schon beendet hat und die Testvorbereitungen treffen kann, erhält die Gruppe QS-SB noch die in der vorherigen Stunde zugesicherte Arbeitszeit. Diese Planung ergab sich aus dem Verlauf der letzten Stunde und aus der Tatsache, daß die nächste Stunde an einem Freitag stattfindet und die Freitagsstunde sich von den zur Verfügung stehenden am besten für die Testarbeiten eignet, so daß der Test heute vollständig vorbereitet werden muß.

Die **gemeinsame** Testplanung ist insofern unverzichtbar, als daß diese ein wichtiges Lernziel für alle ist und auch die Akzeptanz der späteren Testergebnisse auf Seiten der Schüler davon abhängt, inwieweit die Testmodalitäten gemeinsam ausgehandelt sind.

Als alternative Methode in Phase 2 wäre auch eine vorgängige Partnerarbeit mit der Erarbeitung von Vorschlägen zum Test mit anschließender Diskussion im Plenum möglich gewesen. Dann wären allerdings wahrscheinlich die Vorschläge so weit auseinandergegangen, daß ein Kompromiß schwierig und langwierig gewesen wäre. Ich werde also gleich in die Erarbeitung mit einem Gespräch im Plenum einsteigen, da einzelne der zu erarbeitenden Punkte in früheren Stunden von den Schülern schon angerissen worden sind und die Schüler bei dieser Methode die Gelegenheit zum Diskutieren der Testbedingungen nutzen können, was wie gesagt von großer Bedeutung ist. Die Anregung für das Gespräch werde ich also recht offen gestalten ("Was spielt beim Test alles eine Rolle?"). An passender Stelle werde ich die gegebenen Rahmenbedingungen bekanntgeben (zeitlicher Rahmen, vorhandener Test-Text). Da bei **einem** Durchlauf eines Suchverfahrens bei so kurzem Text wie dem verwendeten (ca. 58000 Zeichen) keine vernünftig meßbare Zeit zustande kommt, muß jedes Verfahren mehrfach durchgeführt werden, was ich den Schülern gegebenenfalls noch als zusätzliche Sachinformation im Gespräch geben werde.<sup>33</sup>

Es werden von den Schülern sicher Anforderungen innerhalb der drei Kategorien "Vergleichskriterien", "Anforderungen Text", und "Anforderungen Muster" genannt, da sich diese relativ natürlich ergeben und einzelne Punkte, die unter diese Kategorien fallen, schon von den Schülern angebracht wurden, wie etwa die Anzahl der Vergleiche bzw. die Laufzeit als Vergleichskriterium oder die Frage nach Mustern, die für QS-SB günstig sind. Daher ist diese Kategorisierung sicher sinnvoll.

---

<sup>33</sup> Diese Testmethode wird z.B. auch von Tamm (S. 295) angewandt. Bei von mir durchgeführten Tests mit der Rechneranlage der Schule hat sich eine Durchlaufzahl von 25 als günstig erwiesen, was ich den Schülern an passender Stelle mitteilen werde.

Beim Test von Mustersuchelgorithmen ist es wichtig, Zufallstreffer aufgrund günstiger Muster-Text-Kombination zu vermeiden oder gegebenenfalls offen in die Testplanung insofern einzubeziehen, als daß sie durch entsprechende Mittelwertbildung ausgeglichen werden bzw. man sich ihr bewußt ist;<sup>34</sup>

Beim Testen sollen hier pro Musterlänge drei verschiedene Mustertypen, nämlich ein "normales" und je eines, das für eine der Verbesserungen zweiter Stufe günstig ist, verwendet werden mit anschließender Mittelwertbildung. Dadurch werden die Testergebnisse von vornherein relativiert, so daß den Schülern klar wird, daß es in diesem Bereich ein völlig objektiven Test nicht gibt, da die Frage, was eine typische Mustersuche-Situation in realen Anwendungen ist, nicht geklärt werden kann. Außerdem wird so verhindert, daß die Schüler in der Hausaufgabe durch die Auswahl von Mustern, die auf "ihren" Algorithmus zugeschnitten sind, den Test manipulieren und die Schüler so später die Testergebnisse als nicht regulär ablehnen bzw. anzweifeln. Daher ziehe ich die Alternative, die Schüler ihre Muster ohne Absprachen völlig frei wählen zu lassen, nicht in Betracht.

Ebenso wird die Frage nach der genauen Vorkommensstelle der Muster behandelt: Im Sinne einer Gleichverteilung der **vorkommenden** Muster über den Text wäre es sinnvoll, Muster zu finden, die z.B. genau in der 420. Zeile, aber nicht schon früher auftauchen, was aber recht schwierig sein wird. Diese Situation kann jedoch durch einen Text der Länge 420 und ein darin **nicht vorkommendes** Muster nachgebildet werden. Da uns nur ein einziger Text zur Verfügung steht, der mit etwa 700 Zeilen aber hinreichend lang ist, wird ein kürzerer Text mittels Eingabe der Textlänge durch den Benutzer simuliert, d.h. es wird der komplette Text von der Festplatte gelesen, aber der überschüssige Text wird so von Hand abgeschnitten.

Die Sicherung an der Tafel (s. Tafelbild zur Stunde, S.74) dient dem Festhalten der ausgehandelten Testmodalitäten, insbesondere für die Gruppe QS-D, da der Tafelanschrieb deren Arbeitsauftrag darstellt. Das Tafelbild ist zweigeteilt, auf der linken Hälfte der Tafel stehen jeweils die geforderten Eigenschaften des Tests, auf der rechten Seite die Umsetzung, die nötigen Vorbereitungen etc.

Insgesamt hätte man die Testplanung auch noch umfangreicher gestalten können, man hätte etwa eine Bewertung und Gewichtung von verschiedenen Muster- und Texttypen bezüglich der verschiedenen Algorithmen vornehmen können, um hinterher eine noch breitere Grundlage für die Beurteilung zu haben. Die hier zu erwartenden Ergebnisse sind aber aussagekräftig genug und bieten hinreichend viele Anknüpfungspunkte für eine abschließende Beurteilung der Verfahren.

Eine weitere Aufteilung der Vorbereitungsarbeit auf kleinere Gruppen halte ich nicht für sinnvoll, da die Gruppe QS-D schon bei der Entwicklung ihres

---

<sup>34</sup> vgl. Gronek, S.284

Verfahrens sinnvoll in der ganzen Gruppe gemeinsam an einem Rechner gearbeitet hat.

Jede der Gruppen ist über die Arbeit der jeweils anderen Gruppe einerseits durch die Besprechung in Phase 2, andererseits durch die Präsentation von QS-SB in der letzten Stunde informiert und wird die Ergebnisse im Ausdruck des kompletten Testprogramms Ende der nächsten Stunde erhalten, so daß eine zusätzliche Präsentation der jeweiligen Gruppenergebnisse am Ende dieser Stunde nicht nötig ist.

Das Kopieren der in dieser Stunde erarbeiteten QS-SB-Funktion in die von der Gruppe QS-D in dieser Stunde fertiggestellte Testumgebung und den Einbau der Vergleichszählung in QS-SB werde ich zur nächsten Stunde übernehmen, da ohne dies das Testen nicht stattfinden kann und die Schüler durch die Klausurphase ohnehin stark belastet sind. Damit ist die Testumgebung dann vollständig.

## **7.2 Durchführung**

Wie erwartet heben die Schüler gleich, nachdem der Lehrer das Vorhaben der Stunde erläutert hat, auf die verschiedenen Vergleichskriterien "Laufzeit" und "Vergleichsanzahl" ab, was schon bei der Präsentation von QS-D in der vierten Stunde angedeutet wurde und an der Tafel festgehalten wird. Die entsprechenden Prozeduren zur Laufzeitmessung werden vom Lehrer vorgegeben, und Markus K. bemerkt, daß man zur sichereren Zeitmessung ja auch die Verfahren mehrfach durchführen könne, um Schwankungen auszuschließen, die bei einem einzelnen Durchlauf zu erwarten wären. Der Einbau der Zeitmessung, der Vergleichszählung und entsprechender `For`-Schleifen um die jeweiligen Aufrufe der Suchverfahren wird auf die rechte Seite der Tafel als nötige Vorbereitung notiert. Der L. ergänzt das Argument von Markus K. um die Information, daß für die Messung bei einfachem Durchlauf die Laufzeit zu gering sei.

Als nächstes werden interessanterweise nicht die zu testenden Muster thematisiert, sondern die Texte. Markus R. erläutert, daß man einen normalen Text, einen mit vielen Sonderzeichen, ein PASCAL-Programm und eventuell noch einen mit nur wenigen verschiedenen Zeichen wie etwa beim Problem der Suche in Genketten (s. erste Stunde)<sup>35</sup> nehmen müsse. Dies wird auf der linken Tafelseite notiert, auf der rechten Tafelhälfte jedoch wegen des zeitlichen Rahmen für den Test als nicht realisierbar gekennzeichnet. Der Lehrer gibt hier bekannt, daß zum Testen ein Auszug aus dem Roman Effi Briest zu Verfügung steht, der etwa 700 Zeilen lang ist, und notiert den in einer Unit (s. S.75) bereitgestellten Befehl zum Einlesen des Textes an die entsprechende Stelle auf der Tafel. Stefan bemerkt daraufhin, man müsse aber doch eigentlich unterschiedlich lange Texte haben, woraufhin Dirk die Eingabe der Textlänge von

---

<sup>35</sup> Auch Hume / Sunday untersuchen die Leistungsfähigkeit verschiedener Shift-Tabellen anhand dieses Beispiels (s. Hume / Sunday, S.1246).

Hand durch den Benutzer vorschlägt, was entsprechend fixiert wird. Stefan schlägt als eine mögliche Textlänge zunächst 20 vor, da dies ein hinreichend kurzer Text sei. Die restlichen 680 Zeilen des Textes werden äquidistant in 4 Abschnitte aufgeteilt.

Die bei den Texten aufgezählten Eigenschaften werden jetzt entsprechend auf den Bereich Muster übertragen, wobei besonders der Bereich "Unterschiedliche Typen" diskutiert wird. Jan vertritt die These, daß etwa auch "eyz" ein korrektes Muster der Länge 3 sei (er ist einer der Autoren von QS-SB), was Stefan mit dem Hinweis ablehnt, daß so etwas doch kein Mensch suche und man schon richtige Muster (gemeint sind Wörter) nehmen müsse (er ist einer der Autoren von QS-D). Der Lehrer fragt, welche Musterart denn analog günstig für QS-D sei, woraufhin Eike Muster mit wenig verschiedenen Buchstaben nennt. Als Kompromiß wird schließlich vereinbart, für jede Musterlänge drei sinnvolle Muster entsprechenden Typs zu verwenden und aus den drei jeweils erhaltenen Meßwerten das arithmetische Mittel zu bilden. Nach Tests mit der Rechneranlage im Informatikraum hatte der Lehrer festgestellt, daß jeder Schüler in einer Unterrichtsstunde etwa 2 Musterlängen bearbeiten kann, so daß sich insgesamt 14 Musterlängen ergeben. Festgelegt werden Muster der Länge 4-18. Die Frage nach Vorkommen und Nichtvorkommen wird vom Lehrer mittels der Frage, ob es denn noch andere Gründe für sehr kurze Laufzeit als den **Typ** des Musters gebe, angestoßen. Von Markus K. werden vorkommende und nicht vorkommende Muster genannt, und auf die Frage, was es denn bedeute, daß ein Muster genau z.B. in Zeile 201 auftaucht, schlägt er als mögliche Rationalisierung beim Test vor, dies wie oben angegeben mit der Textlänge zu simulieren, was mittels Durchstreichen des Ausdrucks "vorkommende Muster" an der Tafel festgehalten wird.

Am Ende dieser Phase fragt der Lehrer, was denn wohl später Hausaufgabe sein könnte, woraufhin Dirk erläutert, welche Art von Mustern zu Hause gesucht werden sollen.

Damit sind die Testbedingungen vollständig ausgehandelt.

Die beiden Gruppenarbeiten in Phase 3 verlaufen produktiv:

Die Gruppe QS-SB stellt ihre Lösung am Computer mit Hilfe einer IF-Abfrage, in der vorgängig das seltenste Zeichen von Hand verglichen wird, fertig.

Die Vergleichszählungen innerhalb der Algorithmen machen der Vorbereitungsgruppe leichte Probleme, es wird mit dem Tischmodell jeweils nachkontrolliert. Nach mehreren Versuchen ist die Vergleichszählung, die den schwierigsten Teil der Programmierung ausmacht, fertig, die übrigen Vorbereitungen sind Routinesache. Beim Einbau der Zeitmessung ergibt sich innerhalb der Vorbereitungsgruppe ein interessantes Gespräch:

**Markus K.:** "Sollen wir denn die Vergleichszählung und die Laufzeitmessung in **dem gleichen** Aufruf machen? Die Vergleichszählung kostet doch Zeit!"

*Da keiner aus der Gruppe sich äußert, fragt*

**L.:** "Macht das denn was?"

**Markus K.:** "Es kostet zwar überall die gleiche Zeit, aber man erhält eben nicht die genaue Zeit."

**Dirk:** "Ja, du willst doch auch nur die **Relation** zwischen den Algorithmen bestimmen, also ob einer doppelt so schnell ist wie der andere."

Die übrigen Gruppenmitglieder inklusive Markus K. stimmen Dirk zu und die Zeitmessung wird entsprechend eingebaut.

Am Schluß der Stunde läßt der Lehrer die Hausaufgabe nochmals erläutern, verteilt die Musterlängen auf die Schüler und gibt noch die Test-Formulare aus, in die die Schüler als Vorbereitung für den Test ihre Muster eintragen sollen.

### **7.3 Reflexion**

Insgesamt verlief die Testkonstruktion in Phase 2 zügiger als erwartet, so daß sowohl die Testvorbereitungen als auch QS-SB in Ruhe fertiggestellt werden konnten. Alle für den Test wichtigen Punkte wurden von den Schülern, einmal mit Hilfe eines Hinweises von mir, erarbeitet, so daß der Test in der nächsten Stunde erfolgreich durchgeführt und ausgewertet werden kann.

Das Argument von Markus K., die Algorithmen *im Sinne einer genaueren Messung* mehrfach durchlaufen zu lassen, hatte ich im vorhinein nicht gesehen und war so positiv überrascht.

Bei den Vorschlägen für die Textarten ist besonders der Rückbezug von Markus R. auf eines der Anwendungsbeispiele der ersten Stunde (Basenketten bei der DNS) erfreulich; hierdurch wird klar, daß einerseits die Beispiele bei den Schülern angekommen sind und in die darauf folgenden Überlegungen einbezogen wurden und daß andererseits die in der Textwahl liegende Problematik erkannt wurde.

Ebenso verhält es sich bei den Mustern: Das Gespräch zwischen Jan und Stefan über die Wahl der Muster und der anschließend von allen getroffene Kompromiß zeigen deutlich, daß die Problematik, die sich in der Wahl der Muster verbirgt, erkannt und mittels **bewußter** Auswahl der verschiedenen Mustertypen und Mittelwertbildung gelöst wurde.

Es zeichnet sich hier ab, daß es beim Test und dem Auswerten der Testdaten wohl zu freundschaftlicher Konkurrenz zwischen den beiden Verbesserungen der zweiten Stufe bzw. deren Autoren kommen wird, was ich ebenfalls der kurzen Diskussion zwischen Jan und Stefan entnehme, in der Jan als Co-Autor von QS-SB Muster wählen wollte, die sein Verfahren gut aussehen lassen, was Stefan als Co-Autor von QS-D ablehnte. Die in der Stunde besprochene Wahl der Muster legt diese Manipulation offen und erhöht gleichzeitig die Akzeptanz der Testergebnisse in beiden "Lagern".

In dem interessanten Gespräch zwischen Dirk und Markus K. spiegelt sich ein richtiges Verständnis des Tests wider, der die Bedeutung der Testergebnisse

bezüglich Laufzeit schon gut ausdrückt. Da nach der Äußerung von Dirk das fragliche Problem geklärt war, mußte es nicht mehr unbedingt im Plenum erörtert werden. Falls sich aber in der letzten Stunde der Reihe bei der Auswertung des Tests dazu Gelegenheit ergibt, werde ich die beiden bitten, von dem Gespräch zu berichten.

Insgesamt schienen alle Schüler am Ende der Stunde die getroffene Testplanung verstanden zu haben und sie zu akzeptieren, da es am Ende der Phase 2 und in Phase 4 bei der Zusammenfassung der zu erledigenden Hausaufgaben, die sich ja direkt aus der Testplanung ergeben, weder Fragen noch Widersprüche oder Ergänzungen gab. Dies ist besonders wichtig für die Akzeptanz der Testergebnisse, die die Basis für die abschließende Bewertung der Ideen und Algorithmen in der letzten Stunde darstellen.

## **8 Die sechste Stunde**

In der sechsten Stunde wird der Test gemäß der Planung aus der fünften Stunde von den Schülern durchgeführt. Die Schüler haben beim Ausdenken der Muster viel Phantasie bewiesen, das Testen wird geschickt durchgeführt und macht den Schülern viel Spaß. Sie machen schon beim Notieren der Testdaten erste Beobachtungen (Diskrepanz zwischen Laufzeit und Anzahl der Vergleiche bei QS-D (weniger Vergleiche, aber trotzdem langsamer als QS), sehr hohe Laufzeit bei BF usw.) und suchen nach Erklärungen. Insbesondere die Tatsache, daß das Verfahren QS-SB manchmal mehr Vergleiche benötigt als QS, aber trotzdem schneller ist, ruft Erstaunen hervor, und Markus R. hat schon eine Erklärung, diese wird vom Lehrer aber auf die nächste Stunde vertagt, da erst alle Schüler die Gelegenheit haben sollen, in der Hausaufgabe durch genaue Analyse von QS-SB diesbezüglich Erklärungen dafür zu finden.

Die affektive Bindung der Schüler an die von ihnen entwickelte zweite Verbesserung ist für einen Grundkurs der 12. Jahrgangsstufe erstaunlich: Beim Ablesen der Ergebnisse wird schon freundschaftliche Konkurrenz erkennbar, was durch kommentierende Zwischenrufe an die Schüler mit der jeweils anderen Verbesserung und durch Zurufe an den Computer beim Durchlauf des eigenen Verfahrens ("Los, mach hin") deutlich wird.

Am deutlichsten wird dies am Schluß der Stunde durch die Bemerkung von Jan zu Markus R.: "Hast du gesehen, unseres ist am schnellsten, da bin ich richtig stolz drauf."

Insgesamt hat sich die umfangreiche Testplanung und die Durchführung von Hand durch die Schüler gelohnt. Bei von mir vorgegebener Testplanung oder gar automatisch erstellten Testdaten wären wohl die oben beschriebenen positiven Effekte so nicht zustande gekommen.

Die Schüler erhalten das in dieser Stunde verwendete Testprogramm als Ausdruck und analysieren als Hausaufgabe die von der jeweils anderen Gruppe erarbeiteten Programmzeilen. Diese Hausaufgabe motiviert sich an dieser

Stelle durch die oben beschriebenen Auffälligkeiten bei QS-D und QS-SB, die es in der nächsten Stunde zu präzisieren und zu erklären gilt.

Bei der späteren Durchsicht der Arbeitsergebnisse fielen mir die Messungen bei den Musterlängen 6 und 7 auf, die leicht unter den zu erwartenden Werten lagen (s. S.76). Bei genauerem Hinsehen wurde klar, daß der Schüler, der diese Musterlängen zu bearbeiten hatte, einen kleinen Fehler bei der Testdurchführung gemacht hatte: Er hatte Muster gewählt, die schon früher im Text auftauchen als eigentlich vorgesehen.

## 9 Die siebte Stunde

### 9.1 Planung

#### Thema der Stunde:

Auswertung der Testergebnisse mit abschließender Bewertung der Algorithmen und der zugrunde liegenden Ideen, Ausblick und Abschluß der Reihe

#### Intentionen:

Die Schüler sollen offensichtliche Meßfehler erkennen und benennen, sie sollen die erhaltenen Daten analysieren und Ursachen für die beobachteten Effekte angeben. Sie sollen die erarbeiteten Algorithmen bewerten und Rückschlüsse auf die dahinterliegenden Ideen ziehen; dabei sollen sie ihre Meinung vertreten, Argumente suchen und andere Argumente widerlegen oder anerkennen.

Sie sollen zum Abschluß der Reihe einen Ausblick geben, indem sie weitere mögliche Ideen und Verfahren zur Mustersuche angeben.

#### Geplanter Verlauf:

Unterrichtsphase	Unterrichtsinhalt	Medien	Sozialform Methode
1) Einstieg	Stumme Präsentation der Folie mit dem Testfehler bei Musterlänge 6 und 7. S. äußern sich zunächst spontan.	Folie (s. S.76)	UG
2) Erarbeitung	S. analysieren zu zweit die aus der Teststunde hervorgegangene Tabelle und die Graphen und notieren ihre Beobachtungen in Kurzform auf Folie.	Ausdrucke (s. S.77) PA-Folien	PA
3) Sicherung	Eine Gruppe stellt ihre Ergebnisse vor und begründet sie mit Hilfe der ebenfalls auf Folien vorliegenden Graphen. Die anderen S. können gegebenenfalls nachfragen, widersprechen oder ihre Ergebnisse ergänzen.	PA-Folie Graphen-Folien	SV / UG

4) a) Fazit der Reihe	Ein S. zieht Fazit der Reihe anhand des Ideenbaums. Die Ideen, deren Wirksamkeit und weitere Fortsetzbarkeit werden beurteilt.	Tafel mit schon skizzierten Ideenbaum	SV
b) Ausblick	Es werden weitere "Ideenäste" zur Verbesserung von S. vorgeschlagen (z.B. "Ziehe Vorteile aus den schon erfolgreich ausgegangenen Vergleichen").  S. erhalten als Abschluß die verwendete Testumgebung (Programm und Units) und den c't-Artikel von Tamm über das Verfahren Turbo-Search (s. Literaturverzeichnis) in Form einer HTML-Datei auf Diskette.		

### **Didaktisch-methodische Diskussion:**

Die Messungen der letzten Stunde habe ich zu dieser Stunde in einer Tabelle zusammengefaßt und in graphischer Form aufbereitet (s. S.77ff.). In dieser Stunde sollen die Schüler diese Darstellungen nutzen, um Aussagen über die erarbeiteten Verfahren und Ideen zu treffen, einen Vergleich anzustellen und eine abschließende Beurteilung auszusprechen.

Außerdem soll in dieser Stunde für die Schüler auch Zeit sein, um abschließend noch weitere mögliche Ideen zur Verbesserung, die ihnen "im Kopf herum-schweben", aber eventuell noch nicht ganz ausgereift sind, zu präsentieren.

Zu dieser Stunde habe ich für jeden Schüler die Tabelle mit den zusammengefaßten Testdaten und die Graphen als Ausdrucke vorbereitet. Außerdem liegen alle Graphen auch noch auf Folien vor, so daß die gemachten Beobachtungen bei der Präsentation auch den Mitschülern gut kenntlich gemacht werden können.

Den beobachteten Testdurchführungsfehler möchte ich den Schülern in Form eines stummen Impulses zu Anfang präsentieren. Hier ist auch Gelegenheit für spontane Äußerungen der Schüler zum generellen Ausgang des Tests insgesamt.

Der Arbeitsauftrag zur Phase 2 wird mit der Aufforderung, das Material durchzusehen und alle Beobachtungen zu notieren und eventuell Erklärungen dafür zu suchen, offen gestellt, um auch Vermutungen, Analysen, gegebenenfalls weitere vermutete Fehler u.ä. zuzulassen. Das Datenmaterial und die graphische Aufbereitung sind sehr reichhaltig, so daß viele Erkenntnisse gewonnen werden können.

Für die Analyse der Tabelle und Graphen scheint mir die Sozialform Partnerarbeit sehr gut geeignet, da die Schüler sie gut beherrschen und effizient nutzen. Mir scheint die Gruppenarbeit hier weniger günstig, da die Schüler in Partnerarbeit wohl etwas akribischer nachsehen und argumentieren werden, weil sie stärker persönlich gefordert sind, und so eine größere Vielfalt an Beobachtungen und Argumenten zu erwarten ist.

Eine schriftliche Fixierung der Beobachtungen in Phase 2 ist zur späteren Präsentation nötig und hat zudem noch den Effekt, daß sich die Schüler, um die Folie gut zu füllen, mehreren Effekten zuwenden, anstatt sich an nur einer Sache festzubeißen, wobei dann womöglich noch alle dieselbe Sache betrachten (etwa den Vergleich QS-D und QS-SB), ohne generellere Aussagen zu treffen.

Eine mögliche methodische Alternative in Phase 3 wäre etwa eine Podiumsdiskussion mit vorgegebenen zu vertretenden Meinungen zu den Verfahren oder eine Art Verkaufshow, in der einzelne Schülergruppen je ein Mustersuche-Programm an einen Kunden verkaufen müssen. Vor allem letztere Methode hätte den Vorteil, daß es sicher zu guten Diskussionen unter den Schülern kommen würde. Es waren jedoch schon in der Testdurchführungsstunde gute Ansätze von Beobachtungen und Erkenntnissen da, so daß ich in dieser Stunde die Schüler lieber so frei wie möglich argumentieren lassen möchte. Dafür spricht auch die schon angedeutete Bindung der beiden Gruppen an den "eigenen" Algorithmus und die Tatsache, daß es darum geht, ein ehrliches Fazit zu ziehen und auch generelle Erkenntnisse, die etwa auf jede der Verbesserungen zutreffen (Je länger das Muster, desto kürzer die Laufzeit etc.), zu gewinnen. Da die Diskutierfreude der Schüler gerade beim Thema Mustersuche, wie gesehen, ohnehin hoch ist, rechne ich auch so mit gehaltvollen Diskussionen. Vermutlich wird es ohnehin unter den Schülern eine Spaltung in Befürworter von QS-SB, die etwa Laufzeitvorteile geltend machen, und Gegner, die z.B. die dazu nötige Häufigkeitsverteilung problematisieren, geben.

Bei der Präsentation der Ergebnisse in Phase 3 wird ein zweiter Projektor benötigt, damit die Partnerarbeitsfolie und die jeweilige Graphenfolie gleichzeitig sichtbar sind. Diese Präsentationsphase wird natürlich in eine Gesprächsphase übergehen, wenn die Mitschüler dem Präsentierenden widersprechen, zustimmen oder andere Beobachtungen beisteuern. Hier möchte ich mich möglichst zurückhalten und nur, falls nötig, eingreifen.

Die Bitte um abschließende Beurteilung und Reflexion werde ich bei sich inhaltlich bietender Gelegenheit aussprechen. Der Ideenbaum, der an der hinteren Tafel skizziert ist, kann dabei nochmals Hilfestellung geben. Es sollen mindestens Vermutungen bezüglich QS-Triple, QS-Quattro usw. und bezüglich des Verfahrens SB als direkte Verbesserung von BF (dargestellt durch den grünen Knoten im Baum) angestellt werden. Beides läßt Rückschlüsse auf die Tragfähigkeit der Kopplung bzw. Weiterverfolgung der jeweiligen Ideen zu. Ebenfalls sollen die Ideen von QS und von SB bezüglich ihrer Wirksamkeit verglichen werden.

Bei der abschließenden Ideensammlung sollen die Schüler frei noch weitere Einfälle zur Mustersuche äußern, ohne unbedingt ein ausgereiftes Verfahren oder eine fundierte Argumentation liefern zu müssen. Dies soll dem Ausblick auf das Themengebiet aus Sicht der Schüler dienen. Falls die genannten Ide-

en in der Literatur schon angegeben bzw. untersucht wurden, kann der Lehrer dazu noch Sachinformationen geben. Falls noch Zeit bleibt kann der Lehrer ebenfalls über weitere Ideen berichten.

Als Abrundung und Anregung zum selbständigen Weiterarbeiten gebe ich den Schülern u.a. den c't-Artikel von Tamm als HTML-Dokument auf Diskette mit nach Hause.<sup>36</sup> Einerseits ist die Darstellungsform Hypertext wegen des Internet für die Schüler interessant, andererseits ist die Präsentation auch äußerlich ansprechend gemacht, so daß gerade die Schüler dieser Lerngruppe daran Freude haben werden und einige sich vielleicht den Artikel in den Osterferien ansehen werden.

## **9.2 Durchführung**

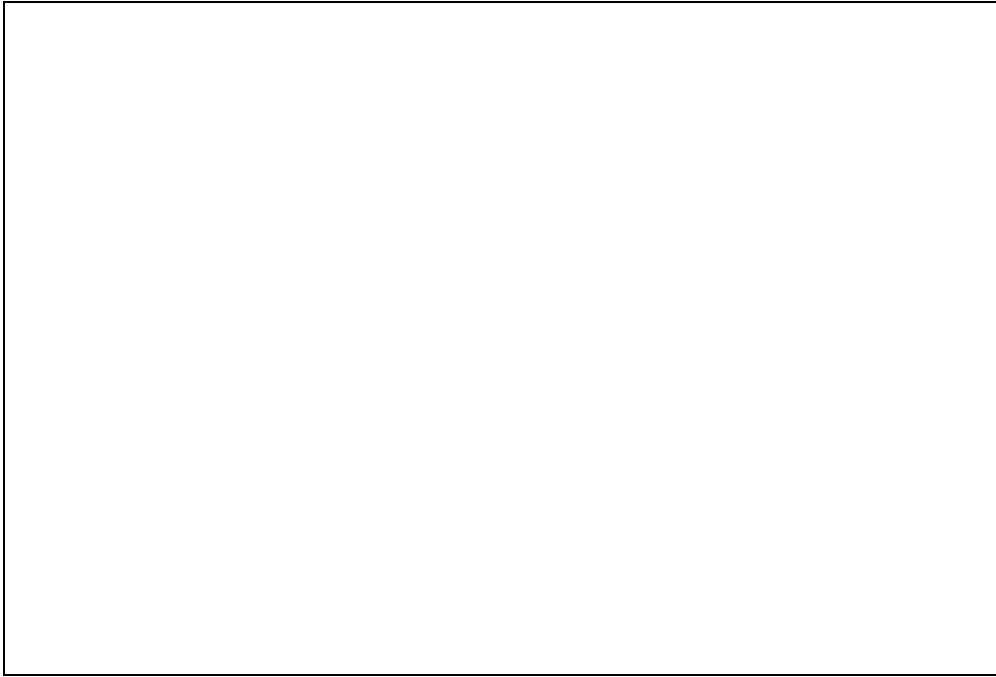
Bei Stundenbeginn blättern die Schüler noch interessiert in ihren Materialien, die umgedreht auf ihren jeweiligen Plätzen liegen. Als der Lehrer den Projektor mit der fehlerhaften Folie anschaltet, schauen die Schüler zunächst nur interessiert und beginnen dann zu kommentieren.

Björn merkt zunächst den großen Laufzeitunterschied zwischen BF und den anderen Verfahren an, Dirk beschwer sich, daß die Darstellung aufgrund des gewählten Maßstabs ungünstig sei: Man könne ja die Verbesserungen überhaupt nicht unterscheiden, woraufhin der Lehrer auf weitere Folien, auf denen die Ergebnisse von BF nicht eingetragen sind und deren y-Achsenausschnitt daher günstiger ist, verweist. Markus R. fragt nun nach den Musterlängen 6 und 7, und "was da wohl passiert" sei. Der Lehrer erläutert die Situation, woraufhin Björn schmunzelnd seinen Fehler eingesteht, da er diese Musterlängen zu bearbeiten hatte.

Der Lehrer erläutert den Arbeitsauftrag für die Partnerarbeit, die Schüler erhalten zu zweit (Markus K./Jan und Stefan/Dirk) bzw. zu dritt (Markus R./Björn/Eike) eine Folie und nehmen ihre Arbeit auf. Während dieser Arbeitsphase ergeben sich innerhalb dieser Gruppen jeweils auch schon Diskussionen über die gemachten Beobachtungen. Dirk merkt an, daß ein prozentualer Vergleich der drei Verbesserungen bezüglich BF oder von QS-D und QS-SB bezüglich QS günstiger gewesen wäre, um die Unterschiede noch deutlicher herausarbeiten zu können.

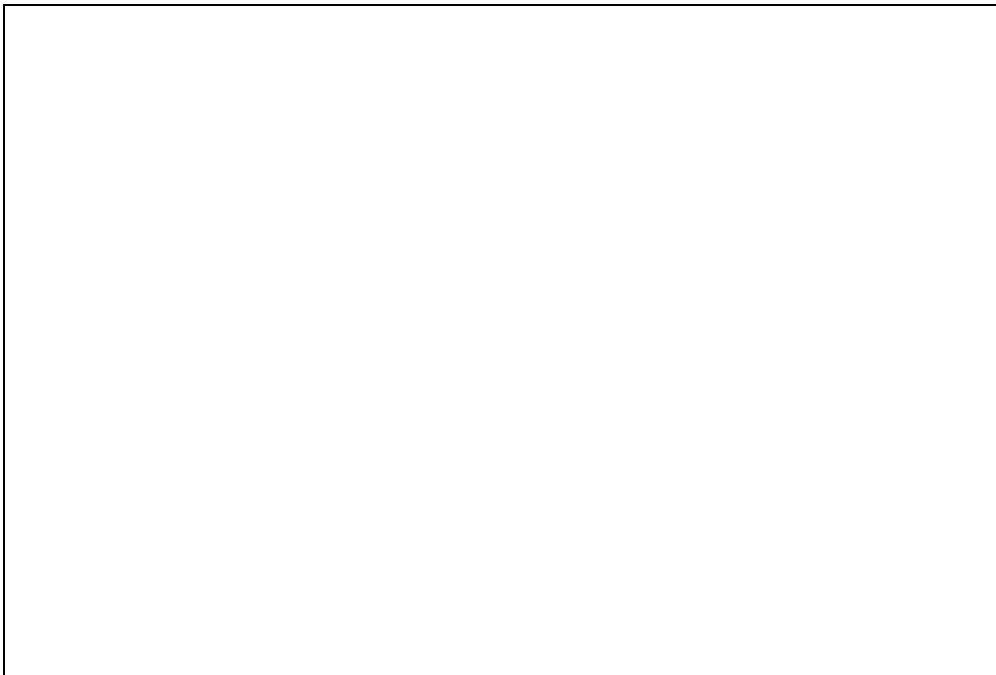
---

<sup>36</sup> Die Genehmigung des Verlags zur Weitergabe dieses Artikels liegt vor.



Stefan und Dirk fixieren ihre Beobachtungen

In Phase 3 erläutert Björn die Erkenntnisse seiner Gruppe, wobei sich recht schnell aus dem Schülervortrag ein Gespräch entwickelt.



Björn erläutert die gemachten Beobachtungen.

Bei **BF** werden folgende Beobachtungen gemacht: Das Verfahren hat immer mit Abstand die meisten Vergleiche, außer bei Musterlänge 4 und die längste Laufzeit; beide Größen bleiben auch im wesentlichen konstant. Bezüglich der Musterlängen kleiner als vier wird vermutet, daß BF, aufgrund der Vorverarbeitung bei den Verbesserungen, wohl bezüglich der Laufzeit, nicht aber bezüglich der Vergleiche günstiger abschneiden wird.

Bei den drei Verbesserungen wird erkannt, daß alle wesentlich günstiger als BF sind und daß bei steigender Musterlänge sowohl die Laufzeiten als auch die Vergleichsanzahlen fallen, was mit den in Mittel größeren Sprungweiten erklärt wird.

Bei **QS-D** fällt den Schülern auf, daß immer etwas weniger Vergleiche gezählt werden als bei QS und QS-SB, aber stets eine höhere Laufzeit zustandekommt, was sie mit der `if`-Abfrage bei der Maximumsauswahl begründen. Weiter wird vermutet daß QS-D bei sehr langen Mustern nicht nur bezüglich der Vergleiche, sondern auch bei der Laufzeit die anderen Verbesserungen schlagen wird, da dann bei der Maximumsauswahl große Unterschiede zustande kommen können, die die `if`-Abfrage aufwiegen (Sachinformation des Lehrers: Etwa ab Musterlänge 60).

Bei **QS-SB** begründet Markus R. die Diskrepanz zwischen der Vergleichsanzahl und der Laufzeit mit der `if`-Abfrage für den Vergleich des seltensten Buchstabens, die der normalen `while`-Schleife vorgelagert ist, und mit seiner Vermutung aus der letzten Stunde, daß oft schon beim ersten Vergleich ein Mismatch auftrete und es dann schneller gehe, eine solche Abfrage zu machen, als für *einen* Vergleich in eine `while`-Schleife einzutreten. Die Aussage, daß QS-SB aber "eigentlich immer" schneller sei, wird von den Schülern der Gruppe QS-D angezweifelt, da bei einer Muster-Text-Kombination, bei der jeweils erst im letzten Muster-Text-Vergleich der Mismatch gefunden würde, ein Vergleich mehr vorgenommen würde, und zwar wegen des doppelten Vergleichs beim seltensten Buchstaben (bei der `if`-Abfrage *und* in der `while`-Schleife). Außerdem seien in den Häufigkeitstabellen ja nur Wahrscheinlichkeiten angegeben, was Markus R. mit dem Hinweis, daß man bei QS-SB "immer mal Pech haben" könne, akzeptiert.

Es gibt energische Auseinandersetzungen zwischen den Gegnern von QS-SB und den Befürwortern, wobei die Frage nach der Gewinnung der Häufigkeitstabellen heftig umstritten ist: Markus K. meint, man müsse pro Text eine neue Tabelle aufstellen und nimmt Bezug auf Anwendungssituationen, wie etwa die Suche in Genketten, in denen es schwierig ist, an eine Tabelle mit allgemein anerkannten Häufigkeiten zu kommen. Jan nimmt eine etwas weitere Sicht ein und erläutert, daß der "Hersteller und Vertreiber des Algorithmus" "für unterschiedliche natürliche Sprachen, Programmiersprachen etc. schon hinreichend viele typische Texte durchsuchen müsse und diese Tabellen im Lieferumfang des Algorithmus bzw. Programms enthalten sein müßten. Wenn man dann die zusätzlichen Vergleiche für das Aufstellen der Tabellen auf "alle in der Welt mit diesem Algorithmus gemachten Mustersuchen" rechne, sei das in Ordnung.

Stefan schlägt vor, für die Häufigkeitstabellen nur Teile des Textes zu analysieren. Auf die Anregung des Lehrers, doch das Suchen und die Erstellung der Häufigkeitstabelle zu koppeln, erläutert er seinen Mitschülern eine adaptive Version von QS-SB, in der es eine feste Starttabelle gibt und die Häufigkeiten

während des Suchens angepaßt werden. Da dies schön zur Phase 4 überleitet und die Argumente im wesentlichen ausgetauscht sind, beendet der Lehrer die Diskussion hier und bittet um abschließende Bemerkungen zum Ideenbaum, der auf der hinteren Tafel skizziert ist und jetzt hervorgeholt wird.

Die Verfahren QS-Tripel bis *QS-Musterlänge* werden erörtert, aber aufgrund der Erfahrung mit QS-Double mit dem Hinweis, daß diese wohl erst bei *sehr* langen Mustern wirklich schneller sind, gleich verworfen. Ebenso wird die Idee "Seltenster Buchstabe" einerseits mit QS-D verknüpft, andererseits auf den von-Hand-Vergleich der seltensten beiden, drei usw. Buchstaben erweitert, was aber auch als zu kompliziert und nicht mehr effizient abgelehnt wird. Auf die Frage, wie denn wohl das "grüne Verfahren", also SB ohne QS abgeschnitten hätte, vermutet es Eike "irgendwo zwischen BF und QS", während Markus R. es "knapp unter BF" platziert, da ja auch QS-SB nur knapp unter QS liege, so daß insgesamt klar wird, daß sich die Einzelentwicklung von SB nicht gelohnt hätte. Jan bemerkt schließlich, daß QS doch ein einfaches und universelles Werkzeug sei, wohingegen QS-D und QS-SB doch wohl "Exoten" seien; das QS-Verfahren solle doch jeder kennen, denn es sei "etwa wie ein Hammer in der Werkzeugkiste".

Beim Ausblick in Phase 4b reagieren die Schüler auf die Frage, was es noch für Ideen bei ihnen gebe, mit Gemurmel, erst auf den Hinweis des Lehrers, daß keine ausgefeilten Algorithmen o.ä. erwartet werden, sondern daß es darum gehe, einfach noch einmal die Phantasie spielen zu lassen, werden mehrere Vorschläge von den Schülern gemacht:

Björn schlägt nun vor, statt einzelner Buchstaben ganze Blöcke fester Länge zu verwenden; Jan möchte die ASCII-Werte der einzelnen Buchstaben des Musters addieren, dasselbe mit dem entsprechenden Textabschnitt tun und die Werte vergleichen.<sup>37</sup> Dirk will aus den erfolgreichen Vergleichen Vorteile ziehen und stellt folgendes fest: Wenn im mittleren Teil des Musters z.B. Buchstaben auftauchen, die vorne nicht zu finden sind, kann man, falls der Mismatch ziemlich weit hinten im Muster auftaucht, ebenfalls Sprünge machen. Daraufhin gibt der Lehrer noch einige Sachinformationen zum KMP-Verfahren, dessen grundlegende Verbesserungsrichtung Dirk beschrieben hat.

Der Lehrer verteilt noch die Disketten, und damit klingt die Stunde aus.

### **9.3 Reflexion**

Insgesamt verlief die Stunde zu meiner vollsten Zufriedenheit: Sowohl die Analyse der Testergebnisse und die Erläuterungen zu den beobachteten Effekten, als auch die abschließende Beurteilung und der von den Schülern gemachte Ausblick waren reichhaltig, gut vorgetragen sowie gehaltvoll, fundiert und mit jeweiligem Bezug auf den Vorredner begründet. Besonders interessant fand ich dabei die Diskussionen um die Behandlung der Häufigkeitsverteilungen, die auch Züge der in der Planung nicht vorgesehenen Ver-

---

<sup>37</sup> Hier finden sich schon Teile der Idee von KR wieder.

kaufsshow hatten. Die zeigt insbesondere Jans Bemerkung über den Vertrieb von Mustersuchelgorithmen und die Umrechnung der Kosten für die Erstellung der Häufigkeitstabellen auf **alle** damit durchgeführten Mustersuchen.

Alle wichtigen Erkenntnisse bezüglich des Vergleichs und der Beurteilung der Algorithmen und Ideen wurden gewonnen und gut begründet.

Die Weiterausnutzung bzw. Kopplung der Ideen wurden sehr schön am Ideenbaum erläutert, ohne daß die Schüler gleich in programmiertechnische Einzelheiten gegangen wären. Diese beiden Heuristiken sind also gut von den Schülern aufgenommen worden.

Insgesamt ergab sich ein sehr schöner Abschluß der Reihe, die "Hammer"-Bemerkung von Jan faßt die Beurteilung gut zusammen.

Die Schüler waren so engagiert, daß es kurz sogar zwei Diskussionsherde gab, etwa bei der Frage nach der Häufigkeitsverteilung für QS-SB. Dies ist auch ein Effekt der positiv zu bewertenden affektiven Bindung der Schüler an "ihr" Verfahren, und die dort erarbeiteten Argumente wurden dann auch dem Plenum vorgetragen.

Die frühe Bemerkung von Dirk bezüglich des relativen Vergleichs wäre vielleicht dann in der Stunde umzusetzen gewesen, wenn keine Folien, sondern der Lehrerrechner mit der Tabellenkalkulation, mit der ich die Auswertung vorgenommen habe, und das OHP-Display benutzt worden wären. Die Folien sind aber leichter für die Schüler zu handhaben, und außerdem sollte das Material, das den Schüler für ihre Analyse zur Verfügung steht, mit dem Präsentationsmaterial übereinstimmen.

Leider konnte in dieser Stunde das Gespräch von Markus K. und Dirk aus der vorletzten Stunde (s. S.52) nicht mehr aufgenommen werden, da dafür die Zeit einfach nicht mehr reichte und ich dies nicht künstlich erzwingen wollte.

Die Stunde hat den Schülern sichtlich Spaß gemacht, was besonders bei den mit Vehemenz, Einsatz und Sachbezug geführten Diskussionen in Phase 3 und Phase 4b erkennbar wurde, in der ein Schüler sogar die Idee von KMP völlig ohne vorherige Kenntnis davon vorschlug.

## **10 Gesamtreflexion**

Die gesamte Reihe war durch große Motivation, Interesse und Spaß auf Seiten der Schüler gekennzeichnet. Es gab Phasen sehr effizienter Schüleraktivität, selbständiger Erarbeitung und vor allem fundierter, sachlicher und engagierter Diskussionen. Die Schüler waren alle gleichmäßig am Unterrichtsgeschehen beteiligt, da für alle Schüler an entsprechenden Stellen Gelegenheit war, sich aktiv in den Unterricht einzubringen.

Die Inhalte sind gut bei den Schülern angekommen. In den späteren Stunden wurde immer wieder Bezug auf die Anwendungsbeispiele genommen, und auch das Argument von Jan bezüglich des weltweiten Vertriebs eines Mustersucheverfahrens (s. S.60) macht die erkannte Bedeutsamkeit sichtbar. Auf die

übrigen sachspezifischen Intentionen bin ich in den Darstellungen der jeweiligen Stunden eingegangen.

Die akribische und detaillierte Planung und Durchführung des Tests und die Tatsache, daß in der letzten Stunde die bisher entwickelten Verfahren konsequent weitergedacht, weitergekoppelt und beurteilt wurden, zeigt, daß die Schüler die fachspezifischen Lernziele (s. S.12) in diesem Themenbereich erreicht haben. Ob die Lernziele auch in der dort formulierten Allgemeinheit erreicht worden sind, nehme ich zwar an, dies kann aber erst bei der späteren Behandlung weiterer Themenbereiche, in denen es ebenfalls um die Verbesserung von Algorithmen unter Kopplung oder Weiterausnutzung von bestimmten Ideen geht, beurteilt werden. Erst dann wird klar, ob diese Heuristiken von den Schülern auf andere Themengebiete übertragen werden können. Ebenso ist natürlich die Testkonstruktion hier exemplarisch zu sehen.

Die Präsentation und Erläuterung eigener Arbeitsergebnisse wurde während der ganzen Reihe hindurch ständig geübt, besonders erfreulich ist hier, daß auch sonst etwas stillere Schüler wie Björn sich hier gut eingebracht haben. Kritik an eigenen Lösungen wurde von den Schülern im kleinen, also bei der Korrektur eigener Fehler oder bei einer besseren programmiersprachlichen Umsetzung etwa von QS, direkt akzeptiert. Generellere Kritik, etwa am Verfahren QS-SB, wurde nach intensiver Diskussion dann auch anerkannt (s. S.60), so daß die sozialen Lernziele als erreicht angesehen werden können.

Der informierende Unterrichtseinstieg in die Reihe war angemessen und die Verbesserung für die Schüler eine so natürliche Sache, daß es keiner weiteren besonderen Motivation bedurfte.

Die Offenheit in der Planung, auch mögliche von Schülern vorgeschlagene Verfahren zu entwickeln und zu implementieren, hat sich in jedem Falle gelohnt. Dadurch konnte der an vielen Stellen zutage tretende positive Konkurrenzeffekt entstehen, der das Erreichen des affektiven Lernziels unterstützte. Dies wurde besonders durch Jans Bemerkung nach der Testdurchführung deutlich ("..., unseres ist am schnellsten, da bin ich richtig stolz drauf", S.54) deutlich. Außerdem wurde dadurch die Reihe sogar auch aus fachlicher Sicht in sinnvoller Weise angereichert (vgl. Reflexion der 2.Stunde, S.27).

Die Sozialformen Partner- bzw. Gruppenarbeit waren oft vorherrschend und für diese Lerngruppe besonders geeignet, was die guten Ergebnisse und der positive Umgang der Schüler miteinander zeigen. Ebenso war das Unterrichtsgespräch hier eine günstige Methode, da die Schüler sich sehr gut beteiligten und, wo immer möglich, eine Schülerdiskussion daraus machten.

Die Medien haben sich hier ebenfalls sehr gut bewährt: Neben den Folien und Arbeitsblättern waren besonders die Tafel- und Tischmodelle einerseits bei den Schülern beliebt, andererseits beim Erkenntnisgewinn sehr hilfreich, was sich auch daraus erkennen läßt, daß die Schüler sie auch bei der selb-

ständigen Gruppenarbeit, wie etwa bei der Entwicklung von QS-D und QS-SB sowie beim Einbau der Vergleichszählung<sup>38</sup>, als Hilfsmittel zur Hand nahmen. Die Notation der Ideen und ihres Zusammenhangs an der Tafel in Form des Ideenbaums war ebenfalls gewinnbringend, was besonders in der fundierten und sicheren Beurteilung der Algorithmen und Ideen in der letzten Stunde der Reihe deutlich wurde.

Die Reihe hat an wenigen Stellen etwas unter den Rahmenbedingungen gelitten (Freitag 6. Stunde, Klausurphase), manchmal war den Schülern am Stundenanfang die Belastung vor allem durch die Klausuren doch ein wenig anzumerken, was aber keine wesentliche Beeinträchtigung darstellte, da sich die Aufmerksamkeit und das Interesse bei den Schülern meist schnell einstellte.

Es gab viele Ansatzpunkte für längere Aktivität bei einzelnen Punkten, wie etwa bei der Frage nach der Mengen-Idee zur Implementation von QS. Bei einer nochmaligen Durchführung der Reihe würde ich den zeitlichen Rahmen etwas ausdehnen, dann wäre etwa auch die Optimierung von BF im kleinen und an einigen Stellen noch größere Eigenständigkeit und Eigenverantwortlichkeit der Schüler möglich.

Die von mir verwendeten nullterminierenden Strings im Rahmenprogramm waren wegen der Numerierung von Null an etwas ungewohnt und bereiteten an einigen Stellen kleine Schwierigkeiten, die durch die Verwendung eines normalen `Array` vom Typ `char` und durch von mir oder gegebenenfalls von Schülern geschriebene Ein- und Ausgabeprozeduren hätte umgangen werden können.

Mir hat die Reihe viel Spaß gemacht, vor allem in Phasen mit hoher Schülereigentätigkeit war Begeisterung für dieses Thema auch bei den Schülern zu spüren. Diese Phasen waren daher auch äußerst produktiv. Das Engagement der Schüler besonders bei der zweiten Verbesserungsstufe war erstaunlich hoch und ist insofern besonders positiv zu werten, als daß die Schüler bei der Diskussion diese Begeisterung auch in die Suche nach rationalen Argumenten für die Überlegenheit ihrer Verbesserung umgemünzt haben. Daher ist die Entscheidung, die Verbesserungen zweiter Stufe arbeitsteilig in Gruppenarbeit machen zu lassen, auch in der Rückschau als sehr günstig zu werten, da nur dadurch die starke affektive Bindung der Schüler, an "ihren" Algorithmus zustande kommen konnte.

Bei der Behandlung der Anwendungsfälle, bei der Frage nach den Häufigkeitstabellen und der Frage, wie man sie gewinnt und wie der dadurch zusätzlich nötige Aufwand zu werten ist (s. S.60), konnte über die reine Algorithmik hinaus begründet diskutiert werden. Die letzte Stunde und besonders Jans

---

<sup>38</sup> s. Foto auf Seite 45

Vergleich von QS mit dem "Hammer in der Werkzeugkiste" (s. S.61) zeigt, daß die Schüler in der Lage sind, abschließende Beurteilungen von Verfahren, die über die rein algorithmische Ebene hinausgehen, vorzunehmen.

Die Tatsache, daß schon in der ersten Stunde die Schüler unaufgefordert Verbesserungsvorschläge gemacht haben, zeigt deutlich, daß sowohl das Themengebiet als auch die Schwerpunktsetzung günstig gewählt waren, da das Themengebiet voll von einfachen und durchschlagenden Ideen ist, mit denen man, auch durch den Ideenbaum unterstützt, gut umgehen kann.

Insgesamt halte ich das Thema vor allem mit dem gewählten Schwerpunkt für sehr geeignet für den Informatikunterricht und würde diese Unterrichtsreihe unter Berücksichtigung der jeweils vorliegenden Lernvoraussetzungen sicher noch einmal durchführen.

## 11 Anhang A: Materialien zur 1. - 4. Stunde

### Tafelbild der 1. Stunde:

<p><b><u>Mustersuche in Texten</u></b></p> <ul style="list-style-type: none"> <li>• Suchen von Basenkettten in der DNS</li> <li>• Textverarbeitung: Wie oft kommt ... im Text vor? Suchen / Suchen-Ersetzen</li> <li>• Internet: Suchmaschinen in bestimmten Datenbanken</li> <li>• Dateisuche im Explorer</li> <li>• Vier gewinnt</li> <li>• Quadratische Wörtersuchrätsel</li> <li>• Bildervergleich</li> <li>• Spracherkennung</li> <li>• Schrifterkennung</li> <li>• Virens Scanner</li> </ul> <p style="text-align: center;">↓</p> <p style="text-align: center;">"Rohtexte"</p> <p style="text-align: center;">Texte ohne Suchstruktur</p>	<p><b><u>Stichwortsuche</u></b></p> <ul style="list-style-type: none"> <li>• Lexikon</li> <li>• Buch mit Index</li> </ul> <p style="text-align: center;">↓</p> <p style="text-align: center;">Texte mit Suchstruktur</p>
--	--

Das erste Beispiel und der erste Teil des zweiten Beispiels zur Mustersuche in Texten waren auf dem Arbeitsblatt 1 als Anregung schon abgedruckt.

$\mathcal{U}$ 

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46			
Benutzen Sie beim Reinigen immer einige Lappen.																																																	
..... einige																																																	
0	1	2	3	4	5																																												

Die Muster in den Tischmodellen waren schon ausgeschnitten und mit Büroklammern an den jeweiligen Text geheftet.

IF g12

## Einfaches Verfahren zur Mustersuche

---

Namen: Markus K. und Jan

Umgangssprachliche Beschreibung des Verfahrens:

Es wird von der 1. Stelle des Textes an der 1. Buchstabe des Musters verglichen. Wird eine Übereinstimmung festgestellt, wird der nächste Buchstabe des Musters überprüft. Ist dies nicht der Fall, wird das Muster um 1. Stelle "weiter geschoben".

Dieses Verfahren wird solange fortgeführt, bis jeder Buchstabe des Musters mit dem entsprechenden des Textes übereinstimmt oder das Muster über den Text "hinausräuft".

## Rahmenprogramm für die Prozeduren zur Mustersuche (1. Stunde):

```

program Mustersuche;
uses strings,crt;

{Die Unit strings liefert Prozeduren und Funktionen zum Umgang mit
langen Zeichenketten. Das sind Arrays vom Typ char, die nicht wie
sonst üblich von Eins, sondern von Null an numeriert werden.}

const Max_Musterlaenge=100;
      Max_Textlaenge=62000; {Diese Größenordnung gibt PASCAL vor}

type tMuster = array [0..Max_Musterlaenge] of char;
     tText  = array [0..Max_Textlaenge] of char;

var Text      : tText;
    Muster    : tMuster;
    Textlaenge,
    Musterlaenge : longint;
    Auswahl    : char;

{////////////////////////////////////Testumgebung////////////////////////////////////}
{////////////////////////////////////Hauptprogramm////////////////////////////////////}

begin {HP}
  repeat
    clrscr;
    writeln(' ');
    writeln(' ');
    writeln(' ');
    repeat Auswahl:=readkey until Auswahl in ['G','g','E','e'];
    clrscr;
    case Auswahl of
      'G','g' : begin readln(Text); Test; end;
      'E','e' : begin clrscr; writeln('Ende des Programms');
                readln; end;
    end; {von case}
    until Auswahl in ['E','e'];
  end.

  {hier kann man die Aufrufe der zu testenden Prozeduren eintragen}
  writeln('Taste drücken!');
  c:=readkey;
  end; {von if}
until StrComp(Muster,'e')=0; {d.h. Muster='e'}
end; {von Test}

{////////////////////////////////////Hauptprogramm////////////////////////////////////}
{////////////////////////////////////Hauptprogramm////////////////////////////////////}

```

## **Umsetzung von QuickSearch in PASCAL**

---

1) Deklariere das besprochene `Array` für die Sprungweiten konkret!

---

2)

a) Mittels welcher Anweisungen wird das `Array` nun mit Werten belegt?  
Welche Probleme ergeben sich dabei? Lösungsvorschlag!

---

---

---

---

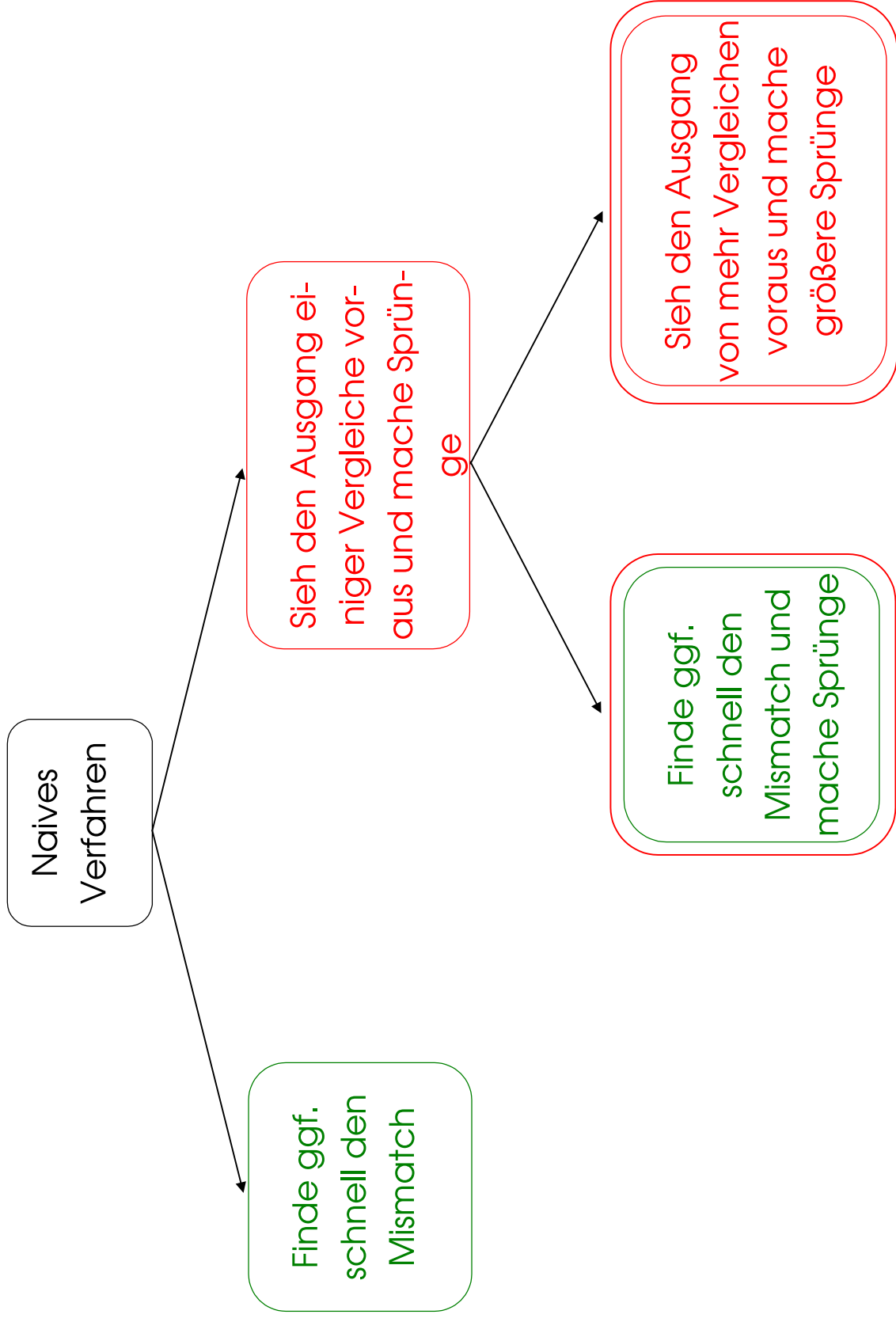
---

b) Was muß man dann an unserer bisherigen Suchprozedur lediglich ändern?

---

---

**Tafelbild der 4. Stunde:**



## Gruppenspezifisches Arbeitsblatt zur 4. Stunde (Gruppe QS-SB):

IF g12

### Weiterentwicklung von QuickSearch

---

Entwickle unsere Version von QuickSearch weiter!

Baue die Idee von Björn und Markus R., zuerst den seltensten Buchstaben, der im Muster vorkommt, zu vergleichen, in die Prozedur ein.

Gehe bei deiner Arbeit davon aus, daß dir eine Häufigkeitstabelle der Form

```
var haeuf : array[0..255] of real;
```

zur Verfügung steht, wobei etwa in `haeuf[32]` die Häufigkeit in Prozent für das 32. ASCII-Zeichen (das ist das Leerzeichen) gespeichert ist.

```
haeuf[0]:=0.00;   haeuf[1]:=0.00;   haeuf[2]:=0.00;   haeuf[3]:=0.00;
haeuf[4]:=0.00;   haeuf[5]:=0.00;   haeuf[6]:=0.00;   haeuf[7]:=0.00;
haeuf[8]:=0.00;   haeuf[9]:=0.00;   haeuf[10]:=1.17;   haeuf[11]:=0.00;
haeuf[12]:=0.00;  haeuf[13]:=1.17;   haeuf[14]:=0.00;   haeuf[15]:=0.00;
haeuf[16]:=0.00;  haeuf[17]:=0.00;   haeuf[18]:=0.00;   haeuf[19]:=0.00;
haeuf[20]:=0.00;  haeuf[21]:=0.00;   haeuf[22]:=0.00;   haeuf[23]:=0.00;
haeuf[24]:=0.00;  haeuf[25]:=0.00;   haeuf[26]:=0.00;   haeuf[27]:=0.00;
haeuf[28]:=0.00;  haeuf[29]:=0.00;   haeuf[30]:=0.00;   haeuf[31]:=0.00;
haeuf[32]:=16.06; haeuf[33]:=0.18;   haeuf[34]:=0.00;   haeuf[35]:=0.00;
haeuf[36]:=0.00;  haeuf[37]:=0.00;   haeuf[38]:=0.00;   haeuf[39]:=0.07;
haeuf[40]:=0.00;  haeuf[41]:=0.00;   haeuf[42]:=0.01;   haeuf[43]:=0.00;
haeuf[44]:=1.84;  haeuf[45]:=0.12;   haeuf[46]:=0.78;   haeuf[47]:=0.00;
haeuf[48]:=0.01;  haeuf[49]:=0.02;   haeuf[50]:=0.01;   haeuf[51]:=0.00;
haeuf[52]:=0.00;  haeuf[53]:=0.00;   haeuf[54]:=0.00;   haeuf[55]:=0.00;
haeuf[56]:=0.00;  haeuf[57]:=0.00;   haeuf[58]:=0.05;   haeuf[59]:=0.05;
haeuf[60]:=0.01;  haeuf[61]:=0.00;   haeuf[62]:=0.01;   haeuf[63]:=0.07;
haeuf[64]:=0.00;  haeuf[65]:=0.30;   haeuf[66]:=0.20;   haeuf[67]:=0.02;
haeuf[68]:=0.26;  haeuf[69]:=0.23;   haeuf[70]:=0.16;   haeuf[71]:=0.26;
haeuf[72]:=0.22;  haeuf[73]:=0.17;   haeuf[74]:=0.06;   haeuf[75]:=0.15;
haeuf[76]:=0.20;  haeuf[77]:=0.22;   haeuf[78]:=0.10;   haeuf[79]:=0.05;
haeuf[80]:=0.08;  haeuf[81]:=0.01;   haeuf[82]:=0.06;   haeuf[83]:=0.42;
haeuf[84]:=0.17;  haeuf[85]:=0.12;   haeuf[86]:=0.11;   haeuf[87]:=0.29;
haeuf[88]:=0.00;  haeuf[89]:=0.00;   haeuf[90]:=0.08;   haeuf[91]:=0.00;
haeuf[92]:=0.00;  haeuf[93]:=0.00;   haeuf[94]:=0.00;   haeuf[95]:=0.00;
haeuf[96]:=0.00;  haeuf[97]:=3.93;   haeuf[98]:=1.24;   haeuf[99]:=2.75;
haeuf[100]:=3.67; haeuf[101]:=12.61; haeuf[102]:=1.03;  haeuf[103]:=1.92;
haeuf[104]:=4.24; haeuf[105]:=6.55;   haeuf[106]:=0.08;  haeuf[107]:=0.72;
haeuf[108]:=2.72; haeuf[109]:=2.21;   haeuf[110]:=8.05;  haeuf[111]:=1.54;
haeuf[112]:=0.25; haeuf[113]:=0.01;   haeuf[114]:=5.05;  haeuf[115]:=4.30;
haeuf[116]:=4.31; haeuf[117]:=2.89;   haeuf[118]:=0.49;  haeuf[119]:=1.22;
haeuf[120]:=0.01; haeuf[121]:=0.01;   haeuf[122]:=0.78;  haeuf[123]:=0.00;
haeuf[124]:=0.00; haeuf[125]:=0.00;   haeuf[126]:=0.00;  haeuf[127]:=0.00;
haeuf[128]:=0.00; haeuf[129]:=0.53;   haeuf[130]:=0.00;  haeuf[131]:=0.00;
```

.....

```
haeuf[253]:=0.00; haeuf[254]:=0.00; haeuf[255]:=0.00;
```

## **Weiterentwicklung von QuickSearch**

---

Entwickle unsere Version von QuickSearch weiter!

Verfolge dazu weiterhin die Idee des Vorausschauens von Vergleichen!

(TIP: Du betrachtest bis jetzt ausschließlich das Textzeichen, das hinter dem Muster kommt. Sieh dir zusätzlich das Textzeichen davor, also das direkt über dem Musterende, an.)



## ***Unit zur Behandlung von langen Strings und zur Zeitmessung:***

```
unit Chr_Help;

{Die zu durchsuchende Textdatei muß zuerst einmal in den
Turbo-Editor geladen und dann unter dem Namen 'Text.dat'
abgespeichert werden.}

interface
uses strings,Uhr1992,crt;

procedure Einlesen(PC:pchar);

procedure Startezeit;
procedure Stoppezeit;
procedure Druckezeit;

{////////////////////}
implementation

procedure Einlesen(PC:pchar);
var Datei : file of char;
    i      : longint;
    c      : char;
begin
    writeln('Datei Text.dat wird geladen');
    assign(Datei,'Text.dat');
    reset(Datei); i:=0;
    while (not eof(Datei)) do
        begin
            read(Datei,c);
            PC[i]:=c;
            inc(i);
        end;
    close(Datei);
end;

writeln('Laden beendet');
end;

procedure Startezeit;
begin
    HoleSystemZeit(Startzeit);
end;

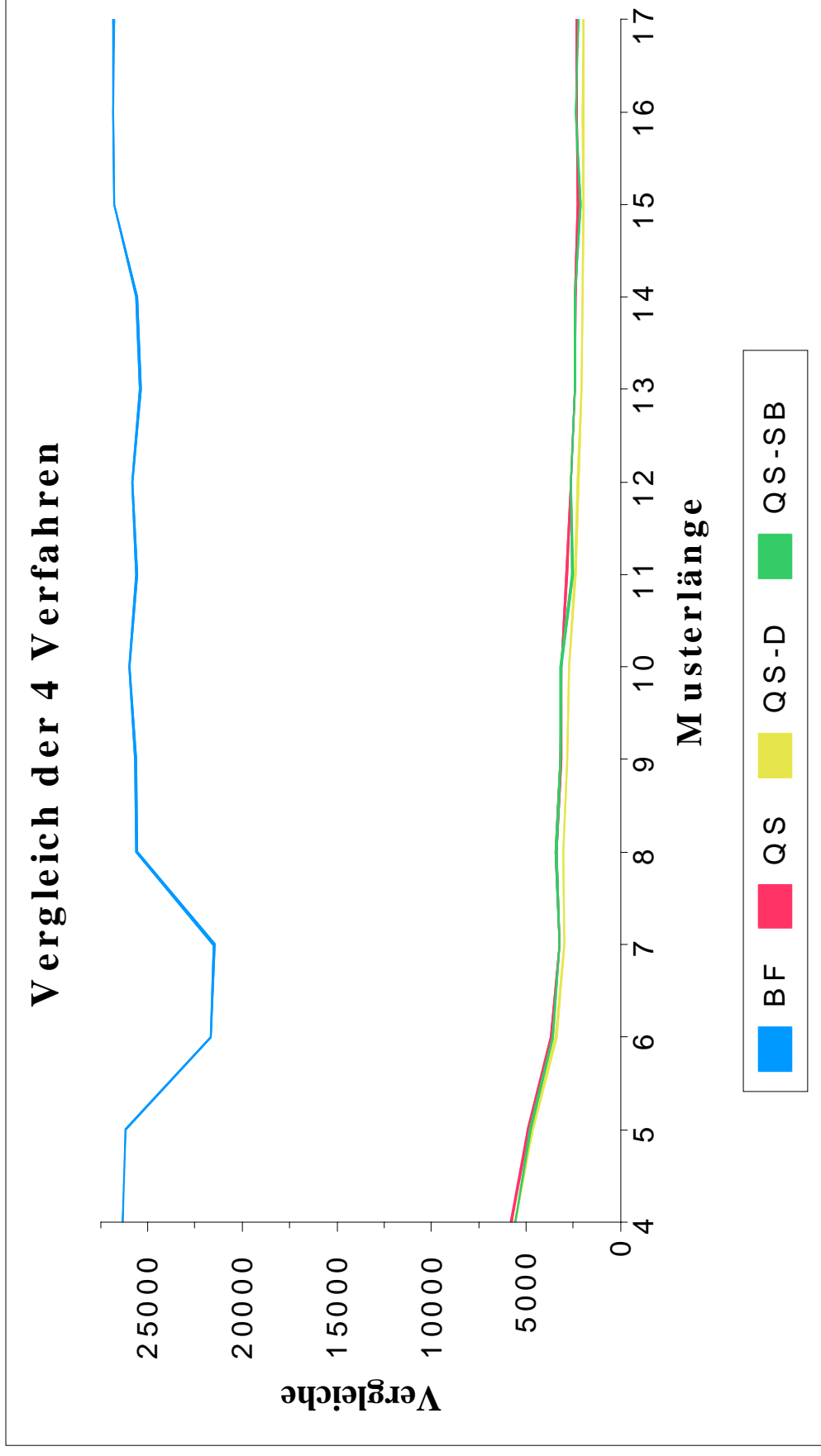
procedure Stoppezeit;
begin
    HoleSystemZeit(Zielzeit)
end;

procedure Druckezeit;
begin
    BerechneLaufzeit(Startzeit,Zielzeit,Laufzeit);
    writeln;
    SchreibeZeit(wherex,wherey,'Benötigte Zeit:',Laufzeit);
    writeln;
end;

{////////////////////}
begin {HP}
end.
```

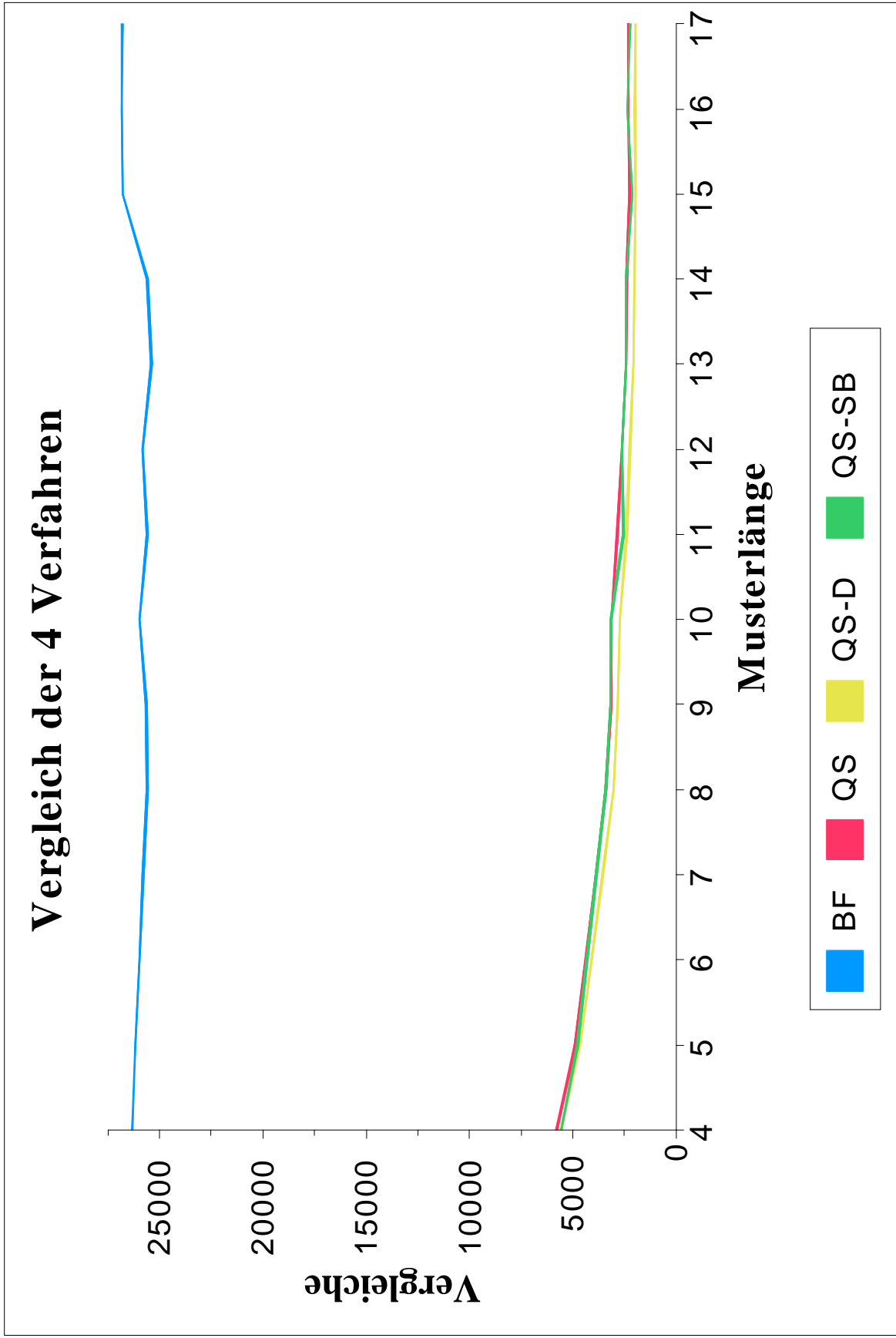
**Darstellung der Meßergebnisse zur 7.Stunde:**

**Vergleichszählungen inklusive des bei Musterlänge 6 und 7 begangenen Testdurchführungsfehlers:**

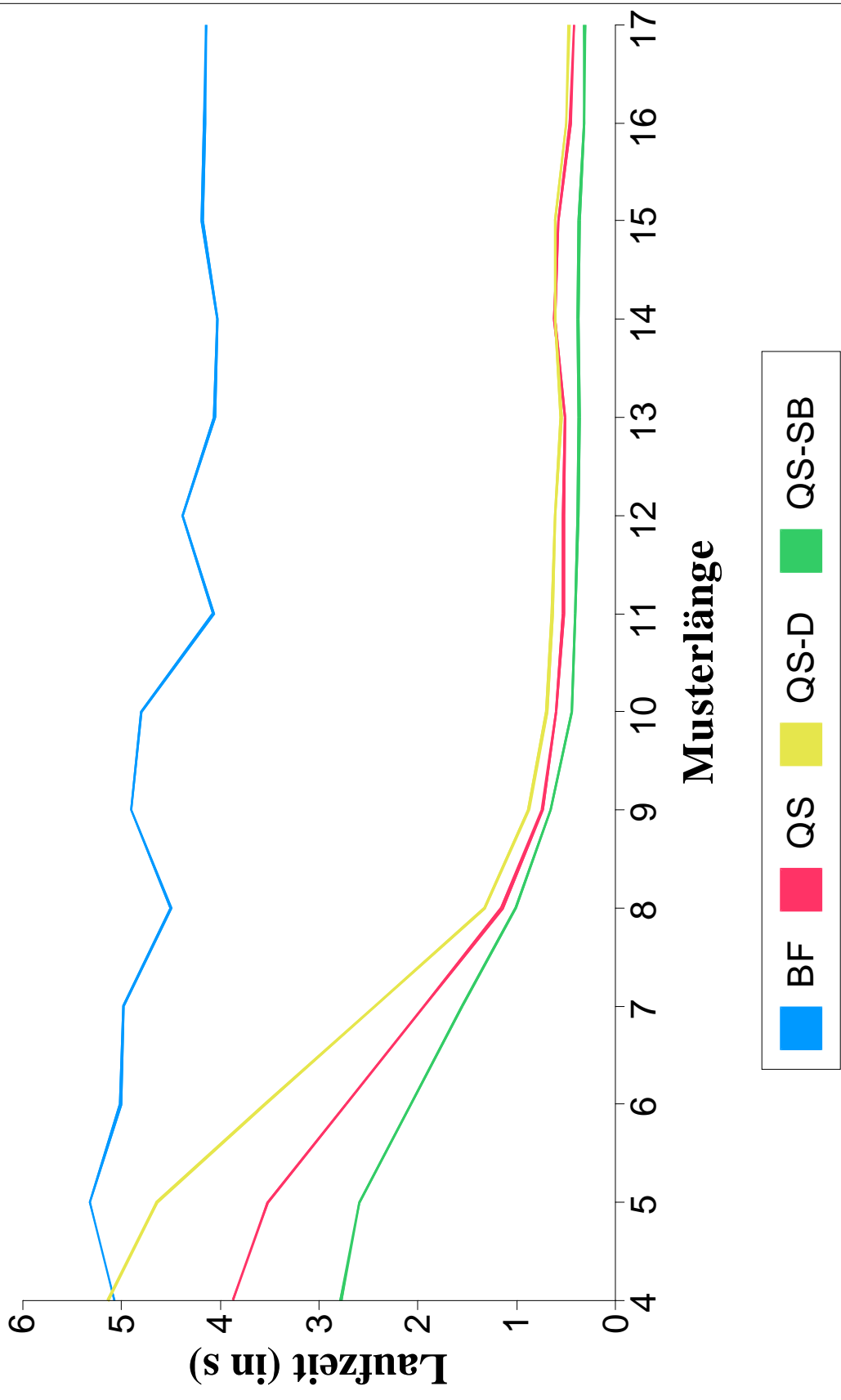


**Tabellarische und graphische Aufbereitung der Meßergebnisse (mit linearer Interpolation bei Musterlänge 6 und 7):**

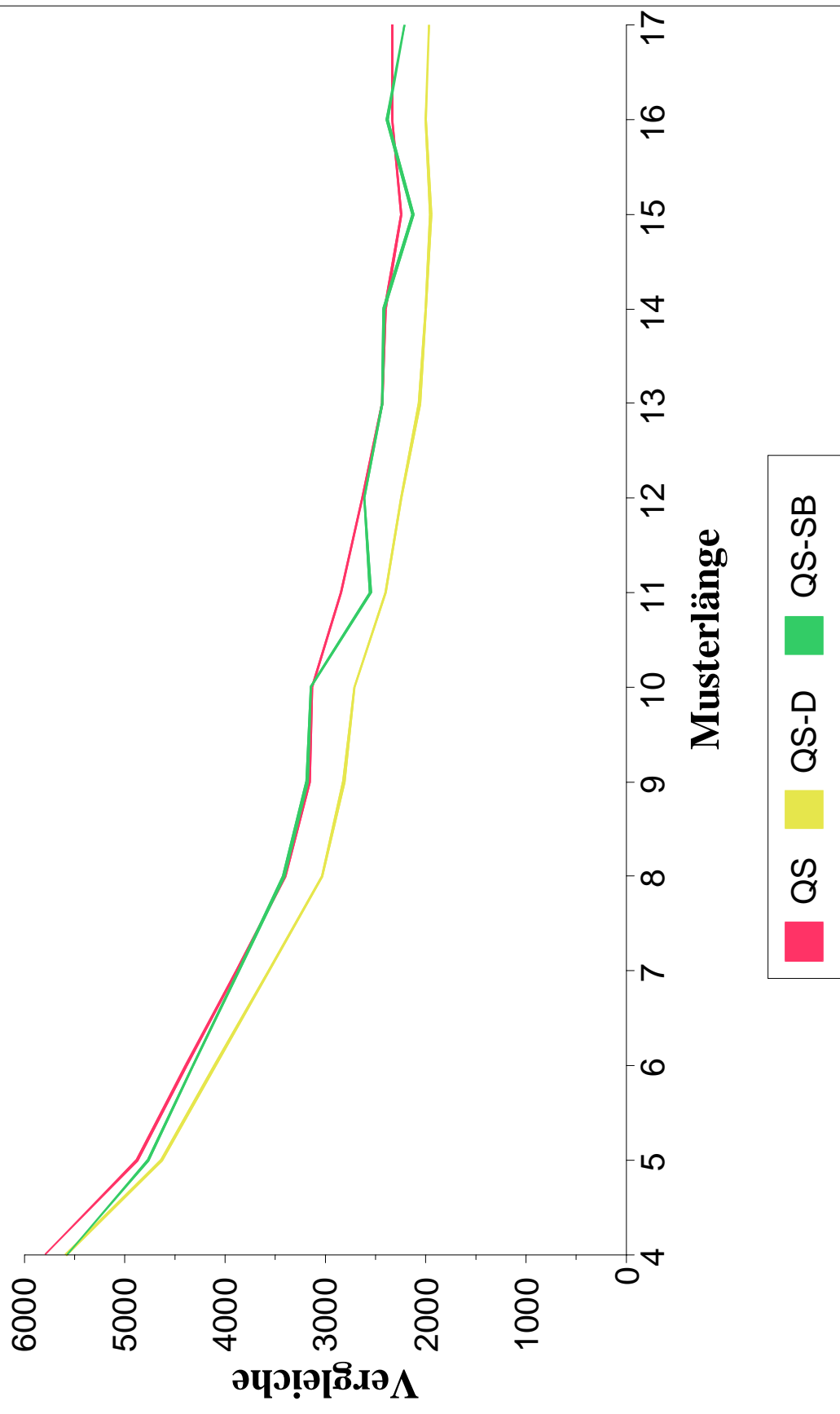
Vergleiche:					Laufzeit:					
Musterlänge:	BF	QS	QS-D	QS-SB		Musterlänge:	BF	QS	QS-D	QS-SB
4	26315	5793	5588	5569		4	5,07	3,87	5,14	2,78
5	26170	4874	4635	4770		5	5,32	3,52	4,64	2,59
6	25973	4383	4101	4319		6	5,01	2,73	3,54	2,07
7	25776	3886	3567	3868		7	4,98	1,94	2,44	1,55
8	25578	3400	3033	3417		8	4,50	1,15	1,33	1,01
9	25632	3158	2818	3184		9	4,90	0,74	0,88	0,66
10	25971	3134	2711	3148		10	4,80	0,60	0,70	0,44
11	25575	2842	2399	2552		11	4,07	0,53	0,64	0,41
12	25803	2633	2249	2612		12	4,38	0,53	0,61	0,38
13	25384	2431	2062	2431		13	4,06	0,51	0,55	0,37
14	25580	2405	2004	2416		14	4,03	0,62	0,61	0,38
15	26767	2246	1952	2129		15	4,18	0,58	0,61	0,37
16	26820	2332	1998	2384		16	4,16	0,46	0,50	0,32
17	26801	2333	1971	2212		17	4,14	0,42	0,47	0,31



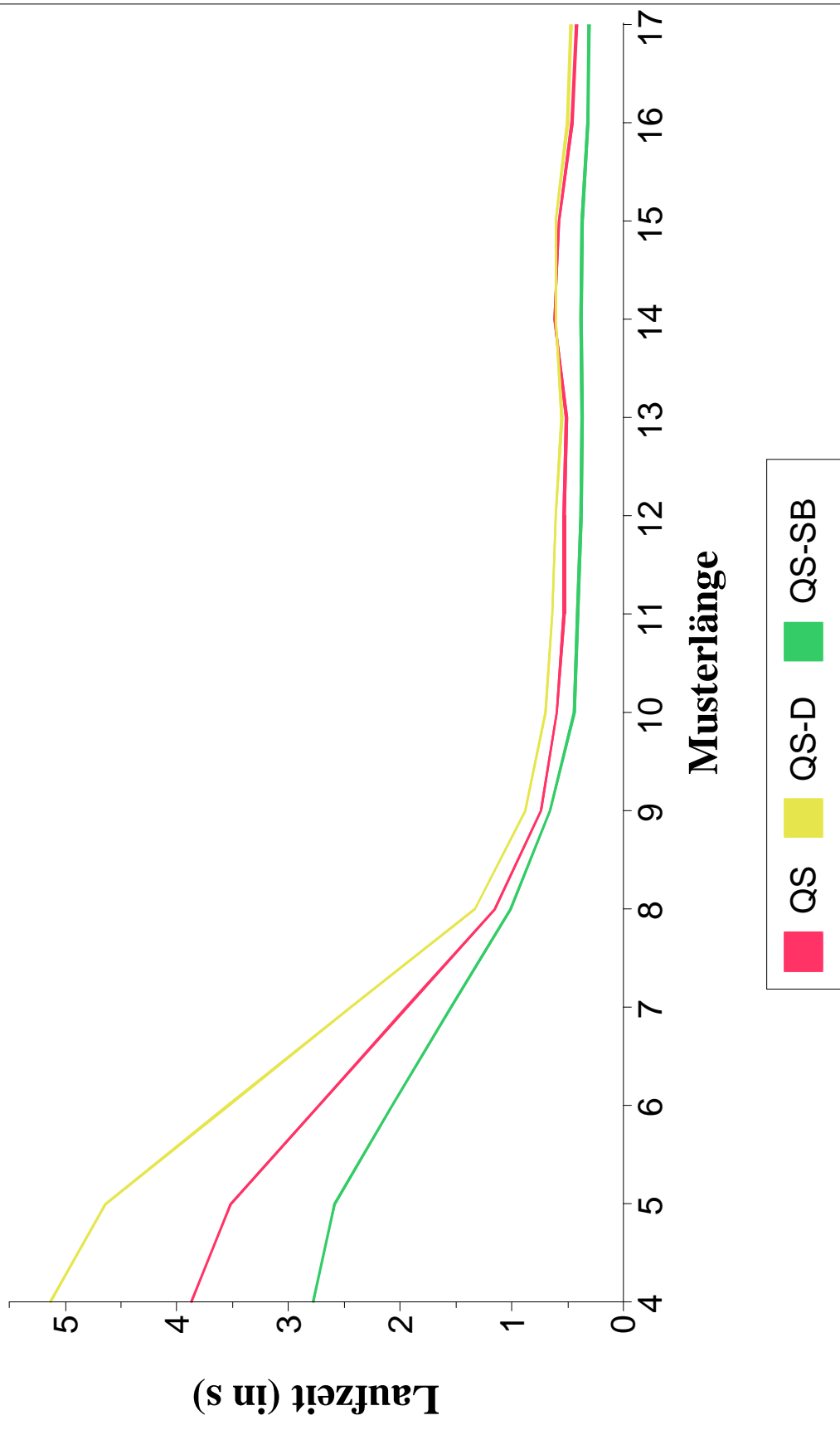
## Vergleich der 4 Verfahren



## Vergleich der 3 Verbesserungen



## Vergleich der 3 Verbesserungen



## 13 Anhang C: Listing des erstellten Programms

```

program Mustersuche;

uses strings, crt, UChrHelp, UHaeuf;

{Die Unit strings liefert Prozeduren und Funktionen zum Umgang mit
langen Zeichenketten. Das sind Arrays vom Typ char, die nicht wie
sonst üblich von Eins, sondern von Null an numeriert werden.}

const Max_Musterlaenge=100;
      Max_Textlaenge=62000; {Diese Größenordnung gibt PASCAL vor.}

type tMuster = array [0..Max_Musterlaenge] of char;
     tText  = array [0..Max_Textlaenge] of char;

var Text      : tText;
    Muster    : tMuster;
    Textlaenge,
    Musterlaenge : longint;
    Auswahl     : char;
    Sprungtabelle : array [char] of longint;
    Vergleiche   : longint;
    Dummy        : longint;

{////////////////////Hilfsprozeduren und -funktionen////////////////}
{////////////////////}
procedure Analyse;
var i : longint;
    c : char;
begin
  for c:=chr(0) to chr(255) do Sprungtabelle[c]:= Musterlaenge+1;
  for i:=0 to Musterlaenge-1 do
    Sprungtabelle[Muster[i]]:=Musterlaenge-i;
  end; {von Analyse}

  inc(i);
  end; {von while}
end; {von BF}

inc(i);
end; {von while}
end; {von BF}

{////////////////////}
function Pos_seltenster : longint;
var max : real;
    i : longint;
begin
  max := 100;
  for i := 0 to Musterlaenge-1 do
    if haeuf[ord(Muster[i])] < max then begin
      max:=haeuf[ord(Muster[i])];
      Pos_seltenster:=i; end;
  end; {von Pos_seltenster}

  {////////////////////Suchfunktionen////////////////}
  {////////////////////}
  function BF : longint;
  var i,j : longint;
      gefunden : boolean;
  begin
    gefunden:=false;
    i:=0;
    j:=0;
    while (i <= Textlaenge-Musterlaenge) and (not gefunden) do
      begin
        j:=0;
        while (j<Musterlaenge) and (Text[j+i] = Muster[j]) do
          begin
            inc(j);
            inc(Vergleiche);
          end; {von while}
          if j=Musterlaenge then begin BF:=i; gefunden:=true; end
          else begin BF:=-1; inc(Vergleiche);
                {wegen der nächsten while-Schleife} end;
        end;
      end;
    end;
  end;

  inc(i);
end; {von while}
end; {von BF}

```

```

{////////////////////////////////////}
function QS : longint;
var i,j      : longint;
    gefunden : boolean;
begin
    Analyse;
    gefunden:=false;
    i:=0;
    j:=0;
    while (i <= Textlaenge - Musterlaenge) and (not gefunden) do
        begin
            j:=0;
            while (j<Musterlaenge) and (Text[j+i] = Muster[j]) do
                begin
                    inc(j);
                    inc(Vergleiche);
                end; {von while}
                if j>=Musterlaenge then begin QS:=i; gefunden:=true; end
            else begin QS:=-1; inc(Vergleiche);
                    {wegen der nächsten while-Schleife} end;
        end; {von QS}

        inc(i,Sprungtabelle[Text[i+Musterlaenge]]);
        end; {von while}
    end;{von QS}

    {////////////////////////////////////}

function QS_D : longint;
var i,j      : longint;
    gefunden : boolean;
begin
    Analyse;
    gefunden:=false;
    i:=0;
    j:=0;
    while (j<Musterlaenge) and (Text[j+i] = Muster[j]) do
        begin
            inc(j);
            inc(Vergleiche);
        end; {von while}
        if j>=Musterlaenge then begin QS:=i; gefunden:=true; end
    else begin QS:=-1; inc(Vergleiche);
            {wegen der nächsten while-Schleife} end;
        end; {von QS}

        inc(i,Sprungtabelle[Text[i+Musterlaenge]]);
        end; {von while}
    end;{von QS}

    {////////////////////////////////////}

function QS : longint;
var i,j, pos      : longint;
    gefunden       : boolean;
    seltenster     : char;
begin
    Analyse;
    gefunden:=false;
    i:=0;
    j:=0;
    pos:=Pos_seltenster;
    seltenster:=Muster[pos];
    QS_SB:=-1;
    while (i <= Textlaenge - Musterlaenge) and (not gefunden) do
        begin
            j:=0;
            inc(Vergleiche); {wegen der if-Abfrage}
            if Text[i+pos] = seltenster then
                begin
                    while (j<Musterlaenge) and (Text[j+i] = Muster[j]) do
                        begin
                            inc(j);
                            inc(Vergleiche);
                        end; {von while}
                        if j>=Musterlaenge then begin QS_SB:=i; gefunden:=true; end
                    else begin QS_SB:=-1; inc(Vergleiche);
                            {wegen der nächsten while-Schleife} end;
                end; {von while}
                inc(i,Sprungtabelle[Text[i+Musterlaenge]])
            end; {von while}
        end; {von QS_SB}

        {////////////////////////////////////}

function QS_SB : longint;
var i,j, pos      : longint;
    gefunden       : boolean;
    seltenster     : char;
begin
    Analyse;
    gefunden:=false;
    i:=0;
    j:=0;
    pos:=Pos_seltenster;
    seltenster:=Muster[pos];
    QS_SB:=-1;
    while (i <= Textlaenge - Musterlaenge) and (not gefunden) do
        begin
            j:=0;
            inc(Vergleiche); {wegen der if-Abfrage}
            if Text[i+pos] = seltenster then
                begin
                    while (j<Musterlaenge) and (Text[j+i] = Muster[j]) do
                        begin
                            inc(j);
                            inc(Vergleiche);
                        end; {von while}
                        if j>=Musterlaenge then begin QS_SB:=i; gefunden:=true; end
                    else begin QS_SB:=-1; inc(Vergleiche);
                            {wegen der nächsten while-Schleife} end;
                end; {von while}
                inc(i,Sprungtabelle[Text[i+Musterlaenge]])
            end; {von while}
        end; {von QS_SB}

        {////////////////////////////////////}

```



## 14 Literaturverzeichnis

Gronek, G.: Optimal schnell, Schnelle Textsuche mit 'Optimal Mismatch', in: c't, Heft 3, 1995: S.278–284

Hume, A.; Sunday, D. M.: Fast String Searching, in: Software — Practice and Experience, Vol. 21(11), November 1991: S.1221-1248

Kultusministerium NRW: Richtlinien für die gymnasiale Oberstufe in Nordrhein-Westfalen Informatik, 1.Auflage 1981, Verlagsgesellschaft Ritterbach, unveränderter Nachdruck, 1988

Landesinstitut für Schule und Weiterbildung, Soest: Gymnasiale Oberstufe: Informatik, Lehrplanentwurf, Stand 10.10.1997

Ottmann, T.; Widmayer, P.: Algorithmen und Datenstrukturen, Mannheim: BI-Wissenschaftsverlag, 1990

Sedgewick, R.: Algorithmen, 2. Auflage, München: Addison-Wesley, 1990

Smith, P.D.: Experiments with a Very Fast Substring Search Algorithm, in: Software — Practice and Experience, Vol. 21(10), October 1991: S.1065-1074

Sunday, D. M.: A Very Fast Substring Search Algorithm, in: Communications of the ACM, Vol. 33, No. 8, August 1990: S.132–142

Tamm, M.: Blitzfindig, Texte schnell suchen mit T-Search, in: c't, Heft 8, 1997: S.292-297

Wirth, N.: Algorithmen und Datenstrukturen mit Modula-2, 4. Auflage, Stuttgart: Teubner, 1986

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt, keine anderen Quellen und Hilfsmittel als die angegebenen benutzt und die Stellen der Arbeit, die anderen Werken dem Wortlaut oder Sinn nach entnommen sind, in jedem einzelnen Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe. Das gleiche gilt auch für beigegebene Zeichnungen, Kartenskizzen und Darstellungen.

Paderborn, den 3.06.1998