

## Übung 2

zu Theoretische Informatik III (für Softwaretechnik)

### Aufgabe 1: dynamisches Programmieren

(9 Punkte)

Gegeben seien  $n$  Matrizen  $A_1 \dots A_n$ , dabei sei  $A_i$  immer vom Typ  $(m_{i-1}, m_i)$  - dies stellt sicher, dass die Matrizen in dieser Reihenfolge multiplizierbar sind. Der Aufwand,  $A_i \cdot A_{i+1}$  zu berechnen, erfordert bei  $m_{i-1} \cdot m_i \cdot m_{i+1}$  Multiplikationen. Es sei das Produkt  $A_1 \cdot \dots \cdot A_n$  zu berechnen. Wie kann man durch Anwendung des Assoziativgesetzes die Anzahl der Multiplikationen minimieren?

Beispiel: Sei  $m_0 := 100, m_1 := 1, m_2 := 100, m_3 := 1$ , dann ist  $A_1$  vom Typ  $(100, 1)$ ,  $A_2$  vom Typ  $(1, 100)$ , und  $A_3$  vom Typ  $(100, 1)$ .

Betrachten wir die möglichen Klammerungen:

$(A_1 \cdot A_2) \cdot A_3$ : Die Berechnung  $B := (A_1 \cdot A_2)$  benötigt  $m_0 \cdot m_1 \cdot m_2 = 10000$  Multiplikationen.  $B$  ist vom Typ  $(m_0, m_2)$ .  $B \cdot A_3$  benötigt  $m_0 \cdot m_2 \cdot m_3 = 10000$  Multiplikationen. Insgesamt sind also 20000 Multiplikationen notwendig. Bei der Klammerung  $A_1 \cdot (A_2 \cdot A_3)$  sind nur 200 Multiplikationen notwendig.

Schreiben Sie ein Programm, welches die optimale Klammerung berechnet und schätzen Sie dessen Laufzeit ab.

### Aufgabe 2:

(5 Punkte)

Gegeben sei

```
function exp(x,y:natural):natural;
begin
  if      y=0          then exp:=1
    else if y mod 2 = 0 then exp:=square(exp(x,y/2))
      else              exp:=square(exp(x,(y-1)/2))*x
  fi fi
end;
```

Die Funktion square quadriert das Argument - führt also eine Multiplikation aus. Was berechnet der Aufruf dieser Funktion? Schätzen Sie deren Laufzeit ab.

**Aufgabe 3:** Teilwörterkennung (nur a) oder b) bearbeiten)

(5 Punkte)

a) Gegeben seien zwei Wörter (strings)  $s_1$  und  $s_2$ . Schreiben Sie einen Algorithmus, der entscheidet, ob  $s_1$  als Teilwort in  $s_2$  enthalten ist, und berechnen Sie den Zeitufwand im average und worst case.

b) In der Literatur befindet sich die Lösung von Boyer-Moore. Erläutern Sie dieses Verfahren allgemein und an einem Beispiel und geben Sie den Laufzeitaufwand an.

**Aufgabe 4:** Greedy

(3 Punkte)

Geben Sie einen Algorithmus an, der zu einem gegebenen Betrag (in Euro und Cent) die minimale Anzahl der Euro und Cent Münzen ausgibt, die zusammen diesen Betrag ergeben.