



Präsenzaufgaben zur Vorlesung »Algorithmen und Datenstrukturen«

Blatt 5

Präsenzaufgabe 5.1 (Laufzeit)

Gegeben sei folgender Programmcode zur Matrixmultiplikation:

```
1 void matmul(int n, int m, int p, double[] a, double[] b, double[] c) {
2     int i, j, k;
3     double sum;
4     for (i = 0; i < n; ++i) {
5         for (j = 0; j < p; ++j) {
6             sum = 0.0;
7             for (k = 0; k < m; ++k) {
8                 sum = sum + a[i + k * n] * b[k + j * m];
9             }
10            c[i + j * n] = sum;
11        }
12    }
13 }
```

Machen Sie sich mit der Funktionsweise des Algorithmus `matmul` vertraut und analysieren Sie den Rechenaufwand anhand der folgenden Tabelle.

Ausdruck	# Operationen
$a \circ b;$	1 $\circ \in \{ +, *, -, /, \&\&, \&, , , \text{not}, <, >, <=, >=, ^ \}$
$a=b;$	1
$i++;$	1
$i = i+1;$	2
$y = x[i];$	2
$x[i] = y;$	2
$x.length;$	1
$A(x);$	$1 + \#A(x)$
$\text{if } B \text{ then}$	$\#B$
$\text{while } B \text{ do}$	$\#B$

Präsenzaufgabe 5.2 (Binomialkoeffizient)

Geben sie ein Programm $\text{Binom}(n, k)$ zur Berechnung des Binomialkoeffizienten $\binom{n}{k}$ mit dynamischer Programmierung in Java oder Pseudocode an.

Präsenzaufgabe 5.3 (Divide and Conquer)

Im Weinkeller eines grausamen Königs befinden sich n wertvolle Weinflaschen. Seine Wächter haben einen Attentäter gefangen genommen, der (genau) eine Flasche vergiftet hat. Dummerweise wissen sie nicht welche. Das Gift ist jedoch so stark, dass man sogar dann sterben würde, wenn man den Wein aller Flaschen vermischt und davon kostet. Allerdings wirkt es so langsam, dass man erst einen Monat später erkrankt. Mit welcher Methode kann der König in einem Monat feststellen, welche Flasche vergiftet ist und dabei höchstens $\mathcal{O}(\log n)$ Vorkoster einsetzen?

Hinweis: Es darf angenommen werden, dass $n = 2^k, k \in \mathbb{N} \cup \{0\}$ Flaschen vorhanden sind.