

Übungsserie Nr. 11

Ausgabe: 13. Mai 2009

Abgabe: 20. Mai 2009

Reversi-Turnier

Das Reversi-Turnier findet am Mittwoch, 27. Mai ab 12:30 Uhr im Raum HG G60 (Semper Aula) statt. Eingeladen dabei zu sein sind selbstverständlich alle! Wer selbst mit einem Programm teilnehmen möchte, muss es bis **Sonntag, 24. Mai 2009** per Mail an kovatsch@inf.ethz.ch einreichen. Den Spieler bitte **unbedingt als Archiv** (zip/tar.gz) inklusive des Package-Verzeichnisses einreichen und als **Betreff: "Reversispieler"** eintragen. Weitere Informationen sind auf der Reversi-Webseite zu finden.

1. Aufgabe: (6 Punkte) Komplexitätsanalyse und O-Notation

(1a) (6 Punkte) Analysieren Sie die sechs folgenden Code-Fragmente in Bezug auf deren Laufzeit und geben Sie das Ergebnis in O -Notation an.

```
// Fragment 1
int diff, sum = 0;
for (int i=0; i<n; i++){
    diff--;
    sum++;
}

// Fragment 2
int sum,a = 0;
for (int i=0; i<n; i++) {
    for (int j=0; j<n; j++)
        sum++;
    a++;
}
```

```

// Fragment 3
int diff, sum = 0;
for (int i=0; i<n*n; i++)
    sum++;
for (int j=0; j<n; j++)
    diff--;

// Fragment 4
int sum = 0;
for (int i=0; i<n; i++)
    for (int j=0; j<n*n; j++)
        sum++;

// Fragment 5
while(n >=1 )
    n = n/2;

// Fragment 6
int sum = 0;
for (int i=0; i<n; i++)
    for (int j=0; j<n*n; j++)
        for (int k=0; k<j; k++)
            sum++;

```

2. Aufgabe: (4 Punkte) Komplexität

(2a) (4 Punkte) Sie führen auf Ihrem Rechner verschiedene Programme aus, deren Laufzeiten in Abhängigkeit von der Grösse der Eingabe hier gegeben sind. Sie können in vorgegebener Zeit eine Eingabe mit einer gewissen Grösse verarbeiten. Sie bekommen nun eine Maschine, die dreimal so schnell arbeitet wie der alte Rechner. Welche Eingabegrösse können Sie nun bearbeiten?

Laufzeit	max. Grösse alt	max. Grösse neu
n	M_1	
n^2	M_2	
2^n	M_3	
$\log n$	M_4	

3. Aufgabe: (2 Punkte) Sortieren mit Suchbäumen

(3a) (2 Punkte) Welche Aussagen über das Sortieren mit binären Suchbäumen sind richtig?

- (i) Im schlechtesten Fall beträgt die Laufzeit $O(n^2)$
- (ii) Im schlechtesten Fall beträgt die Laufzeit $O(n \log n)$
- (iii) Im mittleren Fall beträgt die Laufzeit $O(n \log n)$
- (iv) Im mittleren Fall beträgt die Laufzeit $O(\log n)$

4. Aufgabe: (8 Punkte) Ein Springer auf dem Schachbrett

Betrachten Sie ein Schachbrett (8x8 Felder) und einen einzelnen Springer. Ein Springer bewegt sich immer um zwei Felder horizontal und eines vertikal bzw. ein Feld horizontal und zwei Felder vertikal. Benutzen Sie für die folgenden Aufgaben das Code-Fragment `Springer.java` und die Hilfsklasse `Pos.java`, welche eine Position auf dem Schachbrett repräsentiert. Beide Dateien können Sie sich von der Vorlesungs-Webseite herunterladen.

(4a) (3 Punkte) Gegeben sei eine bestimmte Startposition (x, y) . Welche Felder kann der Springer in maximal n Zügen erreichen? Vervollständigen Sie die *rekursive* Methode `positions()` der Klasse `Springer`, die diese Felder als Menge zurückgibt (und Doppelnennungen vermeidet!).

(4b) (1 Punkt) Welche Positionen können in maximal 2 Schritte ausgehend von der Position $(4, 1)$ erreicht werden? Welche von der Position $(2, 1)$?

(4c) (3 Punkte) Vervollständigen Sie die Methode `path()` der Klasse `Springer`, die zu einer gegebenen Startposition (x, y) einen Weg des Springers über das Brett bestimmt, bei dem jedes Feld des Spielbrettes genau ein Mal besucht wird. Lösen Sie diese Aufgabe mit Backtracking.

(4d) (1 Punkt) Errechnen Sie einen Weg über ein Spielfeld der Grösse 5x5 beginnend bei $(3, 1)$.

Summe: 20 Punkte