Goethe-Universität Frankfurt am Main
Institut für Informatik
Theorie Komplexer Systeme
Dr. Mariano Zelke

24. April 2012

Datenstrukturen

Sommersemester 2012

Übungsblatt 2

Abgabe: *bis 15. Mai 2012, 8.15 Uhr (vor der Vorlesung oder in Raum RM 11-15/113)*

Bitte achten Sie darauf, dass Sie auf der Abgabe Ihrer Lösung Ihren ***Namen****, Ihre* ***Matrikelnummer*** *und Ihre* ***Übungsgruppe*** *angeben. Fehlt eine dieser Angaben, müssen Sie mit* ***Punktabzug*** *rechnen. Mehrseitige Abgaben müssen zusammengeheftet werden.*

Eine Aufgabe gilt nur dann als bearbeitet, wenn neben der Lösung auch die notwendigen Begründungen angegeben sind – es sei denn, in der Aufgabenstellung steht, dass eine solche Begründung nicht erforderlich ist.

Aufgabe 1: (11 Punkte)

(a) Betrachten Sie die folgenden Funktionen von $\mathbb{N}_{>0}$ nach $\mathbb{R}$:

$$\frac{3n}{2},\ 2^{3n},\ n^{3/2},\ \frac{3}{2n},\ \sqrt[3]{n^2},\ 2n^3,\ n^{2^3},\ \log_3 2n,\ 3n^2,\ 3^{2n},\ \frac{3}{2},\ n^{3^2}$$

Ordnen Sie die Funktionen gemäß ihrem asymptotischen Wachstum beginnend mit der am langsamsten wachsenden. Ein Begründung ist nicht erforderlich.

(b) Betrachten Sie die folgenden vier Rekursionsgleichungen. Sie können davon ausgehen, dass n eine Potenz von 4 ist.

(i) $T(1) = 4, \quad T(n) = 4 \cdot T(\frac{n}{2}) + 9n,$ für $n > 1$

(ii) $S(1) = 5, \quad S(n) = 2 \cdot S(\frac{n}{4}) + 3n,$ für $n > 1$

(iii) $R(1) = 6, \quad R(n) = 8 \cdot R(\frac{n}{2}) + 2n^3,$ für $n > 1$

(iv) $Q(1) = 7, \quad Q(n) = Q(n-1) + \frac{3}{2}n,$ für $n > 1$

Bestimmen Sie das asymptotische Wachstum von T, S, R und Q in Θ-Notation. Entscheiden Sie selbst, in welchen Fällen das Mastertheorem angewendet werden kann.

Aufgabe 2: (10 Punkte)

Berechnen Sie die asymptotische Laufzeit der folgenden Pseudocode-Fragmente als Θ-Notation in Abhängigkeit von n. Sie können davon ausgehen, dass n eine Potenz von 4 ist.

(a)

```
int sum=0;
for(i=1; i <= n/2; i++) {
  int k=i;
  sum=sum+k;
  for (j=1; j <= n/4; j++)
    sum=sum-1;
}
```

(b)

```
int sum=10;
for(i=1; i <= n*n; i++)
  for(j=0; j <= n; j++)
    for(k=0; k < j; k++)
      sum=sum+2;
```

(c)
```
while (n >= 1) {
  n=n/4;
  Eine Menge von 5 Anweisungen,
  die n nicht verändern;
}
```

(d)
```
while (n >= 1) {
  Eine Menge von n*n-1 Anweisungen,
  die n nicht verändern;
  n=n/2;
}
```

Aufgabe 3: (9 Punkte)

Es sei ein Array A der Länge n von ganzen Zahlen $(A[1], \ldots, A[n])$ gegeben, wobei $n = 3^k$ gilt für ein $k \in \mathbb{N}$.

(a) Beschreiben Sie zu beiden folgenden Aufgaben je einen Algorithmus in Pseudocode, der die Aufgabe löst. Geben Sie außerdem jeweils die worst-case Laufzeit Ihres Algorithmus als asymptotische Notation in Abhängigkeit von n an.[1]

- **(i)** Der Wert der zweitgrößten Zahl in A soll ausgegeben werden.
- **(ii)** Es sollen zwei Indizes i und j mit $i \neq j$ ausgegeben werden, so dass $A[i]+A[j] = i+j$ ist. Wenn keine zwei solchen Indizes existieren, so soll 0 ausgegeben werden.

(b) Betrachten Sie die folgende rekursive Funktion, die auf dem Array A arbeitet:

```
f(A, int anfang, int ende){
  if (anfang==ende)
    return A[ende];
  else {
    int s = (ende-anfang+1)/3;
    int x = f(A, anfang, anfang+s-1);
    int y = f(A, anfang+s, anfang+2*s-1);
    int z = f(A, anfang+2*s, ende);
    return x+y+z;
  }
}
```

Geben Sie eine Rekursionsgleichung für die Laufzeit $T(n)$ des Funktionsaufrufes `f(A,1,n)` an. Bestimmen Sie die Lösung der Rekursionsgleichung als Θ-Notation.

Aufgabe 4: (6 Punkte)

Alice und Bob spielen (mal wieder) ein Spiel: Alice wählt ein Geheimnis, das Bob mit möglichst wenigen Versuchen erraten soll. In jedem Versuch fragt Bob Alice ein mögliches Geheimnis und sie antwortet nur, ob es sich um das Geheimnis handelt oder nicht. Eine *Strategie* von Bob ist eine Reihenfolge, in der Bob die möglichen Geheimnisse als Anfrage an Alice stellt.
Für die erste Runde verrät Alice, das Geheimnis sei eine Zahl n aus $\mathbb{N}_{>0}$. Die erfolgreiche Strategie von Bob ist zu simpel für eine Frage auf diesem Übungsblatt: Bob rät die Zahlen in $\mathbb{N}_{>0}$ der Größe nach, also in der Reihenfolge 1,2,3,4,5,6, ... und errät das Geheimnis nach n Versuchen.

(a) Für die zweite Runde verkündet Alice, das Geheimnis sei eine Zahl z aus $\mathbb{Z}$. Beschreiben Sie eine Strategie für Bob, z in $\mathcal{O}(|z|)$ Versuchen zu erraten.

(b) Schließlich teilt Alice mit, sie habe ein Paar $(a, b) \in \mathbb{N} \times \mathbb{N}$ gewählt[2]. Existiert eine Strategie für Bob, das von Alice gewählte Paar nach endlich vielen Versuchen zu erraten? Wenn ja, beschreiben Sie eine solche Strategie und geben Sie in asymptotischer Notation an, wie viele Versuche von Bob bei dieser Strategie nötig sind, um das Geheimnis zu erraten.

---

[1] Die Laufzeiten Ihrer Algorithmen müssen nicht optimal sein. Die Anwendung von Sortieralgorithmen ist nicht erlaubt.

[2] Beachten Sie dabei, dass die Reihenfolge von a und b innerhalb des Paares (a, b) eine Rolle spielt; das Paar (1,2) ist ein anderes als das Paar (2,1).