



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2013

25. April 2013

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname: Vorname:

Matrikelnummer: Unterschrift:

Anzahl abgegebener Zusatzblätter:

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.
Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte!).

Die Verwendung von Taschenrechnern, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

| | A1: | A2: | A3: | Summe: |
|---------------------|----------------------|----------------------|----------------------|----------------------|
| Erreichbare Punkte: | 16 | 17 | 17 | 50 |
| Erreichte Punkte: | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Viel Erfolg!

Aufgabe 1.A: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (8 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} n \cdot \log_2(n^{2n}), & \text{wenn } n > 1000 \text{ und ungerade} \\ \sqrt{n^3} + \log_2(2^n), & \text{wenn } n > 300 \text{ und gerade} \\ n^2 \cdot \sqrt{n^3}, & \text{sonst} \end{cases}$$

Kreuzen Sie **in der folgenden Tabelle** die zutreffenden Felder an:

| $f(n)$ ist | $O(\cdot)$ | $\Omega(\cdot)$ | $\Theta(\cdot)$ | keines |
|------------|------------|-----------------|-----------------|--------|
| n^2 | | | | |
| n^3 | | | | |
| $n \log n$ | | | | |
| n | | | | |

Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

b) (8 Punkte)

Tragen Sie für die Codestücke A und B jeweils die Laufzeit in Abhängigkeit von n und die Werte der Variablen a und b nach dem Ausführen des Codes in Θ -Notation **in die unten stehenden Tabellen** ein.

A

```

a = 0;
b = 2;
i = 1;
wiederhole
  b = b + 3;
  für j = 1, ..., [e $\pi$ ] {
    a = a + n;
  }
  i = 2 · i;
bis i ≥ n;
    
```

B

```

a = n;
b = 0;
solange n ≥ 1 {
  a = a · 3;
  für i = 1, ..., n {
    b = b + i;
  }
  n = n - 1;
}
    
```

| Laufzeit | a | b |
|-----------------|-----------------|-----------------|
| $\Theta(\quad)$ | $\Theta(\quad)$ | $\Theta(\quad)$ |

| Laufzeit | a | b |
|-----------------|-----------------|-----------------|
| $\Theta(\quad)$ | $\Theta(\quad)$ | $\Theta(\quad)$ |

Aufgabe 2.A: Sortierverfahren und Datenstrukturen (17 Punkte)

In der Vorlesung haben Sie *Sortieren durch Fachverteilung* (Radix-Sort) kennengelernt. Sei l die maximale Länge der Wörter, m die Größe des Alphabets M und n die Anzahl der Wörter.

- (9 Punkte) Führen Sie auf die folgende *Zahlenfolge* zur Basis 3

$$\langle 21, 122, 211, 1, 12, 121 \rangle$$

den Algorithmus Sortieren durch Fachverteilung aus, um diese aufsteigend zu sortieren. Geben Sie die Inhalte aller Fächer nach jeder Verteilungsphase und die Reihenfolge der Zahlen nach jeder Sammelphase an. Verwenden Sie die dafür vorgefertigten Tabellen.

1. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

1. Sammelphase:

2. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

2. Sammelphase:

3. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

3. Sammelphase:

- (4 Punkte) Geben Sie die Laufzeit von Sortieren durch Fachverteilung in Abhängigkeit von n und l für den besten und schlechtesten Fall in Θ -Notation an und begründen Sie dies kurz.
- (4 Punkte) Was bedeutet es, wenn ein Sortierverfahren *stabil* ist? Ist Sortieren durch Fachverteilung stabil? Begründen Sie Ihre Antwort.

Aufgabe 3.A: Datenstrukturen

(17 Punkte)

Gegeben ist eine doppelt verkettete, azyklische Liste A .

a) (15 Punkte)

Schreiben Sie in detailliertem Pseudocode eine möglichst effiziente Funktion $\text{invert}(A, i, j)$, welche die Reihenfolge der i -ten bis j -ten Elemente der Liste A umkehrt.

b) (2 Punkte)

Geben Sie die Laufzeit Ihres Algorithmus in Θ -Notation an.

Beachten Sie folgende Punkte:

- Der zusätzliche Speicherverbrauch Ihres Algorithmus darf nicht von der Anzahl der Listenelemente abhängen.
- Es soll die Reihenfolge inklusive dem i -ten und j -ten Element umgekehrt werden.
- Die Länge der Liste ($|A|$) kann nur durch Abzählen der Elemente bestimmt werden.
- Sie können davon ausgehen, dass die Liste nicht leer ist und $0 \leq i \leq j \leq |A|$ gilt.
- Der Zeiger A verweist auf das erste Element der Liste A .
- Ein Listenelement a speichert neben verschiedenen anderen Daten jeweils einen Zeiger auf seinen Vorgänger ($a.\text{pred}$) und Nachfolger ($a.\text{next}$). Für das erste bzw. letzte Element gilt $a.\text{pred} = \text{NULL}$ bzw. $a.\text{next} = \text{NULL}$.



186.813 Algorithmen und Datenstrukturen 1 VU 6.0

1. Übungstest SS 2013

25. April 2013

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:

Vorname:

Matrikelnummer:

Unterschrift:

Anzahl abgegebener Zusatzblätter:

Legen Sie während der Prüfung Ihren Ausweis für Studierende vor sich auf das Pult.
Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden. Benutzen Sie bitte dokumentenechte Schreibgeräte (keine Bleistifte!).

Die Verwendung von Taschenrechnern, Mobiltelefonen, PDAs, Digitalkameras, Skripten, Büchern, Mitschriften, Ausarbeitungen oder vergleichbaren Hilfsmitteln ist unzulässig.

| | B1: | B2: | B3: | Summe: |
|---------------------|----------------------|----------------------|----------------------|----------------------|
| Erreichbare Punkte: | 16 | 17 | 17 | 50 |
| Erreichte Punkte: | <input type="text"/> | <input type="text"/> | <input type="text"/> | <input type="text"/> |

Viel Glück!

Aufgabe 1.B: $\Omega/O/\Theta$ -Notation

(16 Punkte)

a) (8 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} n \cdot \log_2(n^{2^n}), & \text{wenn } n > 400 \text{ und ungerade} \\ \sqrt{n^3} + \log_2(2^n), & \text{wenn } n > 1000 \text{ und gerade} \\ n^2 \cdot \sqrt{n^3}, & \text{sonst} \end{cases}$$

Kreuzen Sie **in der folgenden Tabelle** die zutreffenden Felder an:

| $f(n)$ ist | $O(\cdot)$ | $\Omega(\cdot)$ | $\Theta(\cdot)$ | keines |
|------------|------------|-----------------|-----------------|--------|
| n | | | | |
| $n \log n$ | | | | |
| n^3 | | | | |
| n^2 | | | | |

Jede Zeile wird nur dann gewertet, wenn sie vollständig richtig ist.

b) (8 Punkte)

Tragen Sie für die Codestücke A und B in Θ -Notation jeweils die Laufzeit und die Werte der Variablen a und b nach dem Ausführen des Codes **in die unten stehenden Tabellen** ein.

A

```

a = 0;
b = n;
solange n ≥ 1 {
    b = b · 2;
    für i = 1, ..., n {
        a = a + i;
    }
    n = n - 1;
}
    
```

B

```

a = n;
b = 0;
i = 1;
wiederhole
    a = a + 2;
    für j = 1, ..., ⌊eπ⌋ {
        b = b + n;
    }
    i = 2 · i;
bis i ≥ n;
    
```

| Laufzeit | a | b |
|-----------------|-----------------|-----------------|
| $\Theta(\quad)$ | $\Theta(\quad)$ | $\Theta(\quad)$ |

| Laufzeit | a | b |
|-----------------|-----------------|-----------------|
| $\Theta(\quad)$ | $\Theta(\quad)$ | $\Theta(\quad)$ |

Aufgabe 2.B: Sortierverfahren und Datenstrukturen (17 Punkte)

In der Vorlesung haben Sie *Sortieren durch Fachverteilung* (Radix-Sort) kennengelernt. Sei l die maximale Länge der Wörter, m die Größe des Alphabets M und n die Anzahl der Wörter.

- (9 Punkte) Führen Sie auf die folgende *Zahlenfolge* zur Basis 3

$$\langle 20, 201, 2, 21, 122, 221 \rangle$$

den Algorithmus Sortieren durch Fachverteilung aus, um diese aufsteigend zu sortieren. Geben Sie die Inhalte aller Fächer nach jeder Verteilungsphase und die Reihenfolge der Zahlen nach jeder Sammelphase an. Verwenden Sie die dafür vorgefertigten Tabellen.

1. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

1. Sammelphase:

2. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

2. Sammelphase:

3. Verteilungsphase:

| Fach | Datensätze |
|------|------------|
| | |

3. Sammelphase:

- (4 Punkte) Geben Sie die Laufzeit von Sortieren durch Fachverteilung in Abhängigkeit von n und l für den besten und schlechtesten Fall in Θ -Notation an und begründen Sie dies kurz.
- (4 Punkte) Was bedeutet es, wenn ein Sortierverfahren *stabil* ist? Ist Sortieren durch Fachverteilung stabil? Begründen Sie Ihre Antwort.

Aufgabe 3.B: Datenstrukturen

(17 Punkte)

Gegeben ist eine doppelt verkettete, azyklische Liste A .

a) (15 Punkte)

Schreiben Sie in detailliertem Pseudocode eine möglichst effiziente Funktion $\text{invert}(A, i, j)$, welche die Reihenfolge der i -ten bis j -ten Elemente der Liste A umkehrt.

b) (2 Punkte)

Geben Sie die Laufzeit Ihres Algorithmus in Θ -Notation an.

Beachten Sie folgende Punkte:

- Der zusätzliche Speicherverbrauch Ihres Algorithmus darf nicht von der Anzahl der Listenelemente abhängen.
- Es soll die Reihenfolge inklusive dem i -ten und j -ten Element umgekehrt werden.
- Die Länge der Liste ($|A|$) kann nur durch Abzählen der Elemente bestimmt werden.
- Sie können davon ausgehen, dass die Liste nicht leer ist und $0 \leq i \leq j \leq |A|$ gilt.
- Der Zeiger A verweist auf das erste Element der Liste A .
- Ein Listenelement a speichert neben verschiedenen anderen Daten jeweils einen Zeiger auf seinen Vorgänger ($a.\text{pred}$) und Nachfolger ($a.\text{next}$). Für das erste bzw. letzte Element gilt $a.\text{pred} = \text{NULL}$ bzw. $a.\text{next} = \text{NULL}$.