



## 186.172 Algorithmen und Datenstrukturen 1 VL 4.0

### 1. Übungstest SS 2008

25. April 2008

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:  Vorname:

Matrikelnummer:  Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

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

Die Arbeitszeit beträgt 55 Minuten.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	18	16	16	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Erfolg!

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

**(18 Punkte)**

a) (8 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} n \log n + \frac{1}{2}n, & \text{falls } n \text{ prim} \\ n^2 + \frac{1}{3}n, & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$\Omega(\cdot)$	$\Theta(\cdot)$	$O(\cdot)$	keines
$n \log n$				
$n^2$				
$n \log(\log n)$				
$n\sqrt{n}$				

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

b) (4 Punkte)

Welche der Aussagen sind richtig? Kreuzen Sie die korrekten Antworten an.

$$\begin{array}{llll}
 2^n = \Theta(2^{\frac{n}{2}}) & \textcircled{\hspace{0.5cm}} \text{ Ja } \textcircled{\hspace{0.5cm}} \text{ Nein} & \sqrt[5]{n} = \Omega(\log(\log n)) & \textcircled{\hspace{0.5cm}} \text{ Ja } \textcircled{\hspace{0.5cm}} \text{ Nein} \\
 3^n = \Theta(3^{n+5}) & \textcircled{\hspace{0.5cm}} \text{ Ja } \textcircled{\hspace{0.5cm}} \text{ Nein} & \sqrt[4]{n} = O(\sqrt[3]{n}) & \textcircled{\hspace{0.5cm}} \text{ Ja } \textcircled{\hspace{0.5cm}} \text{ Nein}
 \end{array}$$

c) (6 Punkte)

Bestimmen Sie die Laufzeit des unten angegebenen Algorithmus in Abhängigkeit von  $n$  in  $\Theta$ -Notation. Verwenden Sie hierfür einen möglichst einfachen Term.

```

i = 1; c = 0;
solange i < n^2 {
  j = n;
  solange j ≥ 1 {
    für k = 1, ..., j {
      c = c + k;
    }
    j = ⌊ $\frac{j}{3}$ ⌋;
  }
  i = i + 1;
}
    
```

## Aufgabe 2.A: Sortierverfahren

(16 Punkte)

a) (8 Punkte)

Sortieren Sie die nachfolgenden Zahlen **aufsteigend** mittels **Sortieren durch Verschmelzen (Merge-Sort)**:

$\langle 12, 7, 8, 15, 1, 11, 2, 5 \rangle$ .

Geben Sie die Zahlenfolge vor dem ersten und ab dann nach jedem kompletten Verschmelzungsschritt an. Markieren Sie in jeder dieser Folgen die jeweiligen Grenzen zwischen den bereits sortierten Teilfolgen.

Was ist bei der Verschmelzungsphase des Merge-Sort Algorithmus zu beachten, damit dieses Sortierverfahren stabil ist (stabil: die Reihenfolge der Elemente mit gleichem Schlüssel ist vor und nach dem Sortiervorgang identisch)?

b) (8 Punkte)

Gegeben sei der folgende Algorithmus **Sortiere()** in Pseudocode, der für ein Feld  $A = (A[1], \dots, A[n])$  aufgerufen wird:

**Sortiere**(var  $A, n$ )

```
wiederhole {  
    vertauscht = falsch;  
    für  $i = 1, 2, \dots, n - 1$  {  
        falls  $A[i + 1] \geq A[i]$  dann {  
            vertausche  $A[i + 1]$  und  $A[i]$ ;  
            vertauscht = wahr;  
        }  
    }  
} bis vertauscht == falsch;
```

- (5 Punkte) Algorithmus **Sortiere()** arbeitet nicht für jede beliebige Zahlenfolge korrekt. Beschreiben Sie in wenigen Worten, was genau das Problem ist und wie es behoben werden kann.
- (3 Punkte) Bestimmen Sie die Laufzeit im Worst-Case der korrigierten Version von **Sortiere()** in  $\Theta$ -Notation in Abhängigkeit der Eingabegröße  $n$ .

### Aufgabe 3.A: Theorie

(16 Punkte)

a) (4 Punkte)

Für die Analyse von Sortierverfahren sind neben der reinen Laufzeit auch zwei weitere Kenngrößen häufig von großer Bedeutung. Nennen Sie diese beiden Kenngrößen und geben Sie ein Beispiel, warum diese Unterscheidung sinnvoll sein kann.

b) (4 Punkte)

Beschreiben Sie kurz das Prinzip **Teile und Herrsche** (divide and conquer). Nennen Sie zwei beliebige Algorithmen aus der Vorlesung, die auf diesem Prinzip basieren.

c) (4 Punkte)

Eine Folge von Zahlen soll mittels Heapsort **absteigend** sortiert werden. Geben Sie an, ob dafür ein Maximum- oder ein Minimum-Heap nötig ist. Begründen Sie Ihre Antwort.

d) (4 Punkte)

Beschreiben Sie die Grundidee von **Selection-Sort** in 2 bis 3 Sätzen. Nennen sie die Vor- und Nachteile dieses Sortierverfahrens.



## 186.172 Algorithmen und Datenstrukturen 1 VL 4.0

### 1. Übungstest SS 2008

25. April 2008

Machen Sie die folgenden Angaben bitte in deutlicher Blockschrift:

Nachname:  Vorname:

Matrikelnummer:  Studienkennzahl:

Anzahl abgegebener Zusatzblätter:

Legen Sie bitte Ihren Studentenausweis vor sich auf das Pult.

Sie können die Lösungen entweder direkt auf die Angabeblätter oder auf Zusatzblätter schreiben, die Sie auf Wunsch von der Aufsicht erhalten. Es ist nicht zulässig, eventuell mitgebrachtes eigenes Papier zu verwenden.

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

Die Arbeitszeit beträgt 55 Minuten.

	A1:	A2:	A3:	Summe:
Erreichbare Punkte:	18	16	16	50
Erreichte Punkte:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Viel Glück!

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

**(18 Punkte)**

a) (6 Punkte)

Bestimmen Sie die Laufzeit des unten angegebenen Algorithmus in Abhängigkeit von  $n$  in  $\Theta$ -Notation. Verwenden Sie hierfür einen möglichst einfachen Term.

```

i = 1; c = 0;
für i = 1, ..., n {
  j = n;
  solange j ≥ 1 {
    für k = 1, ..., j {
      c = c + k;
    }
    j = ⌊ $\frac{j}{4}$ ⌋;
  }
  i = i + 1;
}
    
```

b) (4 Punkte)

Welche der Aussagen sind richtig? Kreuzen Sie die korrekten Antworten an.

$2^n = \Omega(2^{\frac{n}{2}})$     Ja    Nein    
  $\log(\log n) = O(\log n)$     Ja    Nein  
 $\sqrt[4]{n} = O(\sqrt[5]{n})$     Ja    Nein    
  $4^n = \Theta(4^{n+2})$     Ja    Nein

c) (8 Punkte)

Gegeben sei die folgende Funktion:

$$f(n) = \begin{cases} \frac{1}{3}n^2\sqrt{n} + 5n, & \text{falls } n \text{ prim} \\ n^2 + \log n, & \text{sonst} \end{cases}$$

Kreuzen Sie in der folgenden Tabelle die zutreffenden Felder an:

$f(n)$ ist	$\Omega(\cdot)$	$\Theta(\cdot)$	$O(\cdot)$	keines
$n^2\sqrt{n}$				
$n^2 \log n$				
$n^3$				
$n\sqrt[3]{n}$				

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

## Aufgabe 2.B: Theorie

(16 Punkte)

a) (4 Punkte)

Beschreiben Sie kurz das Prinzip **Teile und Herrsche** (divide and conquer). Nennen Sie zwei beliebige Algorithmen aus der Vorlesung, die auf diesem Prinzip basieren.

b) (4 Punkte)

Für die Analyse von Sortierverfahren sind neben der reinen Laufzeit auch zwei weitere Kenngrößen häufig von großer Bedeutung. Nennen Sie diese beiden Kenngrößen und geben Sie ein Beispiel, warum diese Unterscheidung sinnvoll sein kann.

c) (4 Punkte)

Beschreiben Sie die Grundidee von **Insertion-Sort** in 2 bis 3 Sätzen. Nennen sie die Vor- und Nachteile dieses Sortierverfahrens.

d) (4 Punkte)

Eine Folge von Zahlen soll mittels Heapsort **aufsteigend** sortiert werden. Geben Sie an, ob dafür ein Maximum- oder ein Minimum-Heap nötig ist. Begründen Sie Ihre Antwort.

**Aufgabe 3.B: Sortierverfahren****(16 Punkte)**

a) (8 Punkte)

Gegeben sei der folgende Algorithmus **Sortiere()** in Pseudocode, der für ein Feld  $A = (A[1], \dots, A[n])$  aufgerufen wird:

**Sortiere**(var  $A$ ,  $n$ )

```
wiederhole {  
    vertauscht = falsch;  
    für  $i = 1, 2, \dots, n - 1$  {  
        falls  $A[i + 1] \leq A[i]$  dann {  
            vertausche  $A[i + 1]$  und  $A[i]$ ;  
            vertauscht = wahr;  
        }  
    }  
} bis vertauscht == falsch;
```

- (5 Punkte) Algorithmus **Sortiere()** arbeitet nicht für jede beliebige Zahlenfolge korrekt. Beschreiben Sie in wenigen Worten, was genau das Problem ist und wie es behoben werden kann.
- (3 Punkte) Bestimmen Sie die Laufzeit im Worst-Case der korrigierten Version von **Sortiere()** in  $\Theta$ -Notation in Abhängigkeit der Eingabegröße  $n$ .

b) (8 Punkte)

Sortieren Sie die nachfolgenden Zahlen **absteigend** mittels **Sortieren durch Verschmelzen (Merge-Sort)**:

$$\langle 12, 8, 10, 18, 5, 16, 9, 11 \rangle.$$

Geben Sie die Zahlenfolge vor dem ersten und ab dann nach jedem kompletten Verschmelzungsschritt an. Markieren Sie in jeder dieser Folgen die jeweiligen Grenzen zwischen den bereits sortierten Teilfolgen.

Was ist bei der Verschmelzungsphase des Merge-Sort Algorithmus zu beachten, damit dieses Sortierverfahren stabil ist (stabil: die Reihenfolge der Elemente mit gleichem Schlüssel ist vor und nach dem Sortiervorgang identisch)?