

Norman May

B6, 29, Raum C0.05

68131 Mannheim

Telefon: (0621) 181-2517

Email: norman@pi3.informatik.uni-mannheim.de

Matthias Brantner

B6, 29, Raum C0.05

68131 Mannheim

Telefon: (0621) 181-2517

Email: msb@pi3.informatik.uni-mannheim.de

Algorithmen und Datenstrukturen
Wintersemester 2004/0515. Lösungsvorschlag
25. Februar 2005

Aufgabe 1

15 Punkte

Lösen (Θ -Notation) Sie folgende Rekurrenzen.

Aufgabe 1 a)

5 Punkte

$$T(n) = 3T(n/2) + n$$

Lösung

Anwendung des Mastertheorems

$$a = 3, b = 2, f(n) = n, f(n) = O(n^{\log_b a - \epsilon})$$

Nach dem ersten Fall des Mastertheorems $T(n) = \Theta(n^{\log_b a})$

Aufgabe 1 b)

5 Punkte

$$T(n) = 2T(n/4) + \sqrt{n}$$

Lösung

Anwendung des Mastertheorems

$$a = 2, b = 4, f(n) = \sqrt{n}, f(n) = \Theta(n^{1/2})$$

Nach dem zweiten Fall des Mastertheorems $T(n) = \Theta(\sqrt{n} \lg n)$

Aufgabe 1 c)

5 Punkte

$$T(n) = T(n-1) + 1/n$$

Lösung

Mastertheorem ist nicht anwendbar. Das Aufstellen der Summenformel ergibt 2 Punkte.

Iterationsmethode:

$$\begin{aligned}
 T(n) &= T(n-1) + \frac{1}{n} \\
 &= T(n-1) + \frac{1}{n-1} + \frac{1}{n} \\
 &\vdots \\
 &= T(1) + \sum_{i=2}^n \frac{1}{i} \\
 &= \Theta(1) + (\ln n + O(1) + 1) \\
 &= \Theta(\lg n)
 \end{aligned}$$

Aufgabe 2

7 Punkte

Es soll eine Zahlenfolge A der Länge n aufsteigend sortiert werden. Kreuzen Sie für die verschiedenen Sortierverfahren jeweils *alle* gültigen oberen Schranken für den Sortieraufwand an. Falls Ihrer Meinung nach keine Schranke angegeben werden kann, so kreuzen Sie *keine Angabe möglich* an. Beachten Sie die angegebenen Eigenschaften von A . Gehen Sie von der Laufzeit für den best-case aus.

Aufgabe 2 a)

3 Punkte

Sei A aufsteigend sortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Lösung

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	x	x	x	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	x	x	x	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x

Aufgabe 2 b)

3 Punkte

Sei A absteigend sortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Lösung

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	x	x	x	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x

Aufgabe 2 c)

3 Punkte

Sei A unsortiert.

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Lösung

	$O(\lg n)$	$O(n)$	$O(n \lg n)$	$O(n^2)$	keine Angabe möglich
Insertion-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x	<input type="checkbox"/>
Merge-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Heap-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Quick-Sort	<input type="checkbox"/>	<input type="checkbox"/>	x	x	<input type="checkbox"/>
Bucket-Sort	<input type="checkbox"/>	x	x	x	<input type="checkbox"/>
Counting-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x
Radix-Sort	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	x

Aufgabe 3
5 Punkte

Aufgabe 3 a)
3 Punkte

Fügen Sie die Zahlen 1,7,2,6,3,5,4 in einen leeren Heap ein. Geben Sie nach jedem Einfügen das Heap-Array an.

Lösung

- $\langle 1 \rangle$
- $\langle 7, 1 \rangle$
- $\langle 7, 1 \rangle$
- $\langle 7, 1, 2 \rangle$
- $\langle 7, 6, 2, 1 \rangle$
- $\langle 7, 6, 2, 1, 3 \rangle$
- $\langle 7, 6, 5, 1, 3, 2 \rangle$
- $\langle 7, 6, 5, 1, 3, 2, 4 \rangle$

Aufgabe 3 b)
2 Punkte

Was ist die maximale, was die minimale Anzahl von Elementen in einem Heap der Höhe h ?

Lösung

Maximale Elementanzahl für einen Heap der Höhe h : $2^{h+1} - 1$, minimale Anzahl: 2^h .

Aufgabe 4
12 Punkte

In der Vorlesung wurden die allgemeinen Lösungsstrategien *Greedy*, *Dynamic Programming* und *Divide and Conquer* vorgeschlagen. Wählen Sie zu jedem der folgenden Probleme eine Lösungsstrategie aus, mit der sich das Problem lösen läßt. Geben Sie keine Begründung an!

Aufgabe 4 a)

4 Punkte

Es soll in einem sortierten Integer-Array eine Zahl gesucht werden.

Lösung

Divide-And-Conquer

Aufgabe 4 b)

4 Punkte

Es sollen sehr große Zahlen multipliziert werden. Aufgrund ihrer Größe werden diese als Byte-Arrays mit variabler Länge dargestellt und können daher nicht mittels konstantem Aufwand multipliziert werden. Stattdessen soll der Aufwand einer Multiplikation zweier Zahlen, repräsentiert durch Arrays der Länge n und m , $\Theta(nm)$ betragen. Es gilt nun, für eine gegebene Menge von Zahlen einen vollständig geklammerten Multiplikationsterm mit minimalen Aufwand zu finden, der das Produkt der gegebenen Zahlen zum Ergebnis hat.

Lösung

Dynamisches Programmieren

Aufgabe 4 c)

4 Punkte

Es seien eine Menge von ganzen Zahlen S und eine ganze Zahl M gegeben mit $\forall s \in S : s < M$. Es soll eine Teilmenge S' aus S bestimmt werden, mit $\sum_{s \in S'} s = M$.

Lösung

Dynamische Programmieren

Aufgabe 5

20 Punkte

Aufgabe 5 a)

8 Punkte

Es seien die Sequenzen $X := \langle 3, 2, 1, 4, 7, 6 \rangle$ und $Y := \langle 1, 2, 3, 4, 6, 7 \rangle$ gegeben. Bestimmen Sie eine gemeinsame Teilsequenz mit maximaler Länge für X und Y .

Lösung

Mögliche gemeinsame Teilsequenz von $X := \langle 3, 2, 1, 4, 7, 6 \rangle$ und $Y := \langle 1, 2, 3, 4, 6, 7 \rangle$ mit maximaler Länge:

- $\langle 3, 4, 7 \rangle$
- $\langle 3, 4, 6 \rangle$
- $\langle 2, 4, 7 \rangle$
- $\langle 2, 4, 6 \rangle$
- $\langle 1, 4, 7 \rangle$
- $\langle 1, 4, 6 \rangle$

Aufgabe 5 b)

12 Punkte

Skizzieren Sie einen Algorithmus, der zu einer gegebenen Zahlensequenz der Länge n eine monoton aufsteigende Teilsequenz maximaler Länge bestimmt. Die Laufzeit soll $O(n^2)$ betragen.

Lösung

Die vorangehende Teilaufgabe liefert den Hinweis für den Algorithmus. Dieser besteht aus zwei Schritten. Um für eine Zahlensequenz X der Länge n eine monoton aufsteigende Teilsequenz maximaler Länge zu bestimmen, gilt es zunächst X zu sortieren. Das Ergebnis der Sortierung ist die Sequenz X' . Im zweiten Schritt wird mittels des in der Vorlesung vorgestellten *LCS*-Algorithmus eine gemeinsame Teilsequenz maximaler Länge von X und X' bestimmt. Diese ist die gesuchte monoton aufsteigende Teilsequenz maximaler Länge von X .

Aufgabe 6

7 Punkte

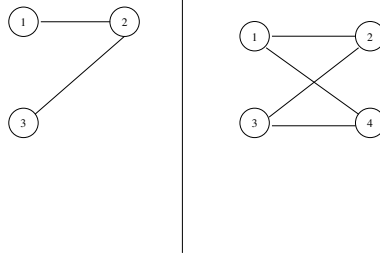
Sei zu $n \in \mathbb{N}$ der ungerichtete Graph $G_n = (V_n, E_n)$ definiert durch: $V_n = \{1, \dots, n\}$ und $E_n = \{(i, j) \mid i, j \in V_n \text{ und } i + j \text{ ungerade}\}$

Aufgabe 6 a)

4 Punkte

Stellen Sie die Graphen G_3 und G_4 graphisch dar.

Lösung



Aufgabe 6 b)

3 Punkte

Beantworten Sie für beliebige $n \in \mathbb{N}$ folgende Fragen. Begründen Sie jeweils kurz Ihre Aussage.

1. Ist G_n vollständig?
2. Ist G_n zusammenhängend?
3. Ist G_n bipartit?

Lösung

1. nein
2. ja
3. ja

Aufgabe 7

7 Punkte

Sei $G = (V, E)$ ein ungerichteter gewichteter Graph und T ein minimaler Spannbaum zu G . Sei $e = (u, v)$ eine Kante mit $u, v \in V$ und dem Gewicht w_e . Geben Sie einen Algorithmus an, der in $O(|V|)$ einen minimalen Spannbaum zu $G' = (V, E \cup \{e\})$ unter Berücksichtigung von T bestimmt. Geben Sie den Algorithmus in Pseudo-Code an.

Lösung

Idee: Pfad zwischen u und v bestimmen. Im Pfad die Kante mit maximalen Gewicht bestimmen. Falls deren Gewicht größer ist als das Gewicht der neu einzufügenden Kante, wird diese Kante im MST T' ersetzt.

MaintainMST($G, T, (u, v, w_e)$)

$G^T := (V, T);$

$T' := T;$

```

BFS( $G^T, u$ );
 $u_m := \pi[v]$ ;
 $v_m := v$ ;
 $w_m := w(\pi[v], v)$ ;
 $c := \pi[v]$ ;
while  $\pi[c] \neq \text{nil}$  do
    if  $w_m \leq w(\pi[c], c)$  then
         $u_m := \pi[c]$ ;
         $v_m := c$ ;
         $w_m := w(\pi[c], c)$ ;
    endif
     $c := \pi[c]$ ;
endwhile
if  $w_e < w_m$  then
     $T' := T' \setminus \{(u_m, v_m, w_m)\} \cup \{(u, v, w_e)\}$ ;
endif
return  $T'$ ;

```