Eidgenössische
Technische Hochschule
Zürich

Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Institut für Theoretische Informatik
Peter Widmayer
Thomas Tschager
Antonis Thomas

16. März 2016

Datenstrukturen & Algorithmen — Blatt 4 — FS 16

Aufgabe 4.1 *Suchbäume.*

a) Zeichnen Sie den entstehenden Baum, wenn die Schlüssel $4, 8, 16, 1, 7, 12, 6, 5$ in dieser Reihenfolge in einen anfangs leeren natürlichen Suchbaum eingefügt werden.

b) Geben Sie die Preorder-, Postorder- und Inorder-Reihenfolgen für den Baum aus a) an.

c) Löschen Sie aus dem Baum aus a) zunächst den Schlüssel 1 und danach aus dem entstehenden Baum den Schlüssel 8. Zeichnen Sie beide Bäume.

d) Zeichnen Sie das Ergebnis, wenn die Schlüssel aus Aufgabenteil a) in einen anfangs leeren AVL-Baum eingefügt werden.

Aufgabe 4.2 *Sortieralgorithmen.*

Beantworten Sie die folgenden Fragen und begründen Sie kurz Ihre Antwort.

a) Ist die sortierte Folge $1, 2, \ldots, n$ ein Min-Heap?

b) Wenn alle Zahlen in einem Min-Heap unterschiedlich sind, an welchen Positionen kann dann das grösste Element stehen?

c) Ein vergleichsbasiertes Sortierverfahren heisst *stabil*, wenn die relative Reihenfolge identischer Elemente nicht verändert wird. Ein Sortierverfahren heisst *in-situ*, falls es ausser der Eingabefolge mit konstantem Speicherplatz für die Folgenelemente auskommt. Welche der Ihnen bekannten Sortierverfahren sind stabil und welche in-situ, bzw. einfach entsprechend anpassbar?

Bitte wenden.

Aufgabe 4.3 *Zweidimensionales Maximum Subarray Problem (**Programmieraufgabe**).*

Gegeben sei eine ganzzahlige $(n \times n)$-Matrix A. Entwerfen und implementieren Sie einen Algorithmus, der eine Teilmatrix berechnet, deren Summe der Einträge maximal ist. Eine $(a \times b)$-Teilmatrix einer $(n \times n)$-Matrix ist hier definiert als die Matrix, die entsteht, wenn ein zusammenhängender $(a \times b)$-Block der Einträge von A betrachtet wird $(0 \leq a, b \leq n)$.

Eingabe Die erste Zeile der Eingabe enthält lediglich die Anzahl der Testfälle t. Danach folgen t Testfälle, die wie folgt repräsentiert werden: Die erste Zeile enthält die Dimension der Matrix $n \leq 100$. Danach folgen n Zeilen mit n ganzzahligen Einträgen, die A darstellen. Dabei entspricht die i-te Zeile exakt der i-ten Zeile der Matrix A, d.h. sie enthält die Einträge $a_{ij} \leq 100$ für $1 \leq j \leq n$.

Ausgabe Für jeden Testfall soll eine Zeile ausgegeben werden, die die grösste Summe der Einträge einer Teilmatrix enthält.

Beispiel

Eingabe:

```
3
2
-1 3
3 -1
2
-2 -3
-1 -4
2
2 -1
-2 -1
```

Ausgabe:

```
4
0
2
```

Bemerkungen Wie beim zweiten Testfall oben gesehen werden kann, ist auch die leere (0×0)-Teilmatrix eine gültige Lösung. Der Judge überprüft drei Kategorien von Testfällen:

- **Easy** (20 Punkte): Für diese Testfälle gilt $n \leq 20$.
- **Medium** (30 Punkte): Für diese Testfälle gilt $n \leq 50$.
- **Hard** (50 Punkte): Für diese Testfälle gilt $n \leq 100$.

Hinweise: Die drei Kategorien von Testfällen können mit verschiedenen Ansätzen gelöst werden. Während die leichten Testfälle durch einen Bruteforce-Algorithmus mit Laufzeit $O(n^6)$ gelöst werden können, benötigen die Testfälle der zweiten Kategorie bereits einen effizienteren Algorithmus mit einer Laufzeit von $O(n^4)$, der Teilsummen vorberechnet. Für die dritte Kategorie muss das Problem auf das Maximum Subarray Sum-Problem, das in der Vorlesung vorgestellt wurde, reduziert werden. Das führt zu einem Algorithmus mit Laufzeit $O(n^3)$. Natürlich löst ein Algorithmus für eine Kategorie auch die Testfälle aller vorherigen Kategorien.

Abgabe: Am Mittwoch, den 23. März 2016 in Ihrer Übungsgruppe.