

Zusatzaufgaben 03

Algorithmen I – Sommersemester 2022

Gesamtpunkte: 35

Aufgabe 1 - Kleinaufgaben (4 Punkte)

1. Ordne die folgenden Algorithmen absteigend nach ihrer Worst-Case Laufzeitkomplexität: Quicksort, Bucketsort, Mergesort (1 Punkt)
2. Wie lautet die untere Schranke für die Worst-Case Laufzeitkomplexität bei vergleichsbasiertem Sortieren? Gilt diese untere Schranke auch für Sortieralgorithmen für ganze Zahlen? Begründe. (1 Punkt)
3. Was macht ein stabiles Sortierverfahren aus? Nenne ein Beispiel für ein stabiles Sortierverfahren. (1 Punkt)
4. Nenne die Heap-Eigenschaft bezüglich eines Arrays $A[0, \dots, n-1]$, welches implizit einen binären Heap darstellt. (1 Punkt)

Aufgabe 2 - Pseudocode (8 Punkte)

```
FIRST( $A: [\mathbb{N}; n], x: \mathbb{N}, y: \mathbb{N}, z: \mathbb{N}$ )
  if  $x < y$  then
     $u: \mathbb{N} = \text{SECOND}(A, x, y, z)$ 
     $z_1: \mathbb{N} = x + (z \bmod (u - x))$ 
     $z_2: \mathbb{N} = u + 1 + (z \bmod (y - u - 1))$ 
    FIRST( $A, x, u - 1, z_1$ )
    if  $u < y$  then
      FIRST( $A, u + 1, y, z_2$ )
    end
  end
```

```
SECOND( $A: [\mathbb{N}; n], x: \mathbb{N}, y: \mathbb{N}, z: \mathbb{N}$ ):  $\mathbb{N}$ 
```

```
   $u: \mathbb{N} = A[z]$ 
```

```
  SWAP( $A[y], A[z]$ )
```

```
   $i: \mathbb{N} = x$ 
```

```
   $j: \mathbb{N} = x$ 
```

```
  while  $j < y$  do
```

```
    if  $A[j] \leq u$  then
```

```
      SWAP( $A[i], A[j]$ )
```

```
       $i := i + 1$ 
```

```
    end
```

```
     $j := j + 1$ 
```

```
  end
```

```
  SWAP( $A[i], A[y]$ )
```

```
  return  $i$ 
```

1. Seien $A = \langle 4, 8, 12, 34, 19, 26 \rangle$ und $n = 6$. Gib die den Zustand von A nach einem Aufruf von FIRST mit Eingabe $(A, 0, n - 1, 1)$ an. (1 Punkt)
2. Was berechnet FIRST für eine beliebiges Array A , was berechnet SECOND? Haben wir diesen Algorithmus schon in der Vorlesung kennengelernt? Wie war sein Name? (2 Punkte)
3. Ändere den Pseudocode von FIRST bzw. SECOND so ab, dass er stabil wird, sich die Laufzeit aber nicht ändert. Du darfst dabei die Funktion $\text{CONCAT}(A1, A2)$ benutzen um zwei Arrays in Linearzeit zu konkatenieren. (3 Punkte)
4. Der Algorithmus FIRST ist in-place, das heißt er benötigt nur konstant viel zusätzlichen Speicherplatz. Wie viel zusätzlichen Speicherplatz braucht dein Algorithmus? Gib im O-Kalkül an. (1 Punkt)
5. Nenne einen Algorithmus der das gleiche tut wie FIRST, aber stabil und in-place ist. Was ist seine Laufzeit in O-Kalkül? (1 Punkt)

Aufgabe 3 - Min-Heaps (11 Punkte)

1. Folgende Arrays stellen Max-Heaps dar. Falls die Heap-Eigenschaft verletzt wurde, gib an wo, wenn nicht, zeichne die Baumrepräsentation des

Heaps. (2 Punkte)

- a) $\langle 56, 47, 56, 10, 20, 50, 51, 1, 2, 3, 18 \rangle$
- b) $\langle 56, 56, 47, 10, 20, 50, 51, 1, 2, 3, 18 \rangle$
- c) $\langle 56, 47, 56, 10, 51, 20, 50, 1, 2, 3, 18 \rangle$
- d) $\langle 56, 47, 56, 10, 5, 20, 50, 1, 2, 3, 18 \rangle$

2. Erstelle aus Array $A = \langle 7, 8, 1, 2, 6, 7, 5 \rangle$ einen binären Min-Heap mit BUILD aus der Vorlesung. Gib dabei das Array an, sobald sich Einträge darin ändern. Du musst nur die vertauschten Zahlen angeben, bei allen anderen wird angenommen, dass sie gleich bleiben. (2 Punkte)
3. Füge in den Min-Heap $A = \langle 2, 7, 3, 9, 7, 5, 5 \rangle$ mit Hilfe von PUSH zuerst ein Element mit Priorität 4 und anschließend ein Element mit Priorität 1 ein. Gib deinen Heap nach jeder Einfügeoperation an. (2 Punkte)
4. Lösche aus dem Min-Heap $A = \langle 2, 7, 3, 9, 7, 5, 5 \rangle$ mit Hilfe von POPMIN zwei Mal das kleinste Element. Gib das Array nach jeder Operation an. (2 Punkte)
5. Man kann ein Array A sortieren, indem man mit BUILD einen Heap aus A ausbaut und mit POPMIN wiederholt das Minimum entfernt. Dieses Sortierverfahren heißt HEAPSORT und ist im Allgemeinen nicht stabil.
 - a) Gib ein Beispiel an, bei dem die Stabilität eines Arrays verloren geht. (1 Punkt)
 - b) Beschreibe, wie man ohne die Laufzeit oder Funktionsweise von HEAPSORT zu verändern Stabilität erzwingen kann. Du hast dazu $\mathcal{O}(n)$ zusätzlichen Speicher gegeben. (2 Punkte)

Aufgabe 4 - Weltraum-Scanner (9 Punkte)

Das ALMA (Atacama Large Millimeter Array) ist eines der größten Radioteleskope der Erde. Das Teleskop besteht aus 66 Antennen, welche variabel angeordnet werden können, um verschiedene Zoomgrößen und Ausschnitte des Himmels zu erreichen. Durch geschickte mathematische und algorithmische Berechnungen erzeugt das Teleskop so ein Bild der Größe $N \times N$ wobei N von der aktuellen Konfiguration des Teleskops abhängt.

Im Folgenden wollen wir neu gefundene Planetensysteme in einer Galaxie indexieren. Pro aufgenommenem Bild erhalten wir dazu maximal G Koordinaten $(x_i, y_i) \in [0, N - 1] \times [0, N - 1]$, an denen Planetensysteme erkannt wurden.

1. Da sich die Erde dreht und stets nur ein Teil des Himmels gleichzeitig aufgenommen werden kann, gibt es Überschneidungen der Bilder. Wir wollen nun auf die Planetensysteme aus einem gegebenen Abschnitt der Form $[w_1, w_2] \times [0, N - 1]$ oder $[0, N - 1] \times [h_1, h_2]$ zugreifen. Beschreibe, wie du die gegebenen Daten so abspeichern kannst, dass in Zeit $\mathcal{O}(\log(G) + k)$ auf einen Abschnitt mit k Planetensystemen zugegriffen werden kann. Begründe deine Antwort. (3 Punkte)
2. Aufgrund bisheriger Erfahrungswerte gehen wir davon aus, dass pro Bild $\log(G)$ neue Planetensysteme entdeckt werden. Wir können in $\mathcal{O}(\log(G))$ überprüfen, ob eine Planetensystem schon entdeckt wurde. Um ein neues Planetensystem zu registrieren und zusätzliche Daten zu speichern benötigen wir $\mathcal{O}(G^3)$ Zeit, da eine Spektralanalyse gemacht werden muss. Zeige, dass ein beliebiges Planetensystem in amortisiert $\mathcal{O}(G^2 \log(G))$ Zeit abgearbeitet wird. (2 Punkte)
3. Für jedes Planetensystem erstellt die Spektralanalyse einen (a,b)-Baum, in welchem die Planeten des Planetensystems nach dem Abstand zu ihrem Stern sortiert sind.

Im Folgenden betrachten wir einen Himmelsabschnitt mit k Planetensystemen. Beschreibe einen Algorithmus, der in $\mathcal{O}(G^3 \log(G) + k)$ alle Planeten aus der Habitablen Zone des Himmelsabschnitts zurückgibt. Du darfst dabei annehmen, dass ein Planetensystem niemals mehr als P Planeten hat, da wir noch kein Planetensystem mit mehr Planeten gefunden haben. Die Habitable Zone H_i ist für jedes Planetensystem G_i unterschiedlich und wurde ebenfalls während der Spektralanalyse bestimmt. Die Habitable Zone ist der Bereich, in dem Leben existieren kann. Sie wird repräsentiert durch ein Tupel aus minimalem und maximalem Abstand zum Stern des Planetensystems. (4 Punkte)

Aufgabe 5 - To negative infinity and beyond! (3 Punkte)

In der Vorlesung wurde der ∞ -Trick für (a,b) -Bäume vorgestellt. Beim $-\infty$ -Trick hat das Dummy-Element den Wert $-\infty$. Wie unterscheidet sich ein (a, b) -Baum, bei dem der $-\infty$ -Trick angewendet wird von einem, bei dem der ∞ -Trick angewendet wird? Beachte insbesondere Veränderungen in den Operationen, die wir in der Vorlesung kennengelernt haben. (3 Punkte)