Institut für Theoretische Informatik
T.T.-Prof. Dr. Thomas Bläsius
https://scale.iti.kit.edu

Übungsblatt 4

Algorithmen 1 - Sommersemester 2023

Abgabe im ILIAS bis 19.05.2023, 18:00 Uhr

Die Abgabe erfolgt als *eine* PDF-Datei über das Übungsmodul in der Gruppe deines Tutoriums im ILIAS.
Beachte bitte die Hinweise zum Bearbeiten auf der Webseite (neu).

Gesamtpunktzahl: 20

Aufgabe 1 – Geheimrezept (8 Punkte)

Wir betrachten die folgenden drei Algorithmen in Pseudocode. Beachte, dass wir dabei häufig über Mengen iterieren, bei denen eine implizite Ordnung angenommen wird.

```
HYDROGEN(A : [ℕ; n]):
 1  for i ∈ {0, 1, ..., n − 1} do
 2      j_m := i
 3      m := A[j_m]
 4      for j ∈ {i + 1, i + 2, ..., n − 1} do
 5          if A[j] > m then
 6              j_m := j
 7              m := A[j_m]
 8          end
 9      end
10      for j ∈ {j_m − 1, j_m − 2, ..., 0} do
11          A[j + 1] := A[j]
12      end
13      A[0] := m
14  end
```

```
HELIUM(A : [ℕ; n], k : ℕ):
1  c : ℕ = 0
2  for i ∈ {0, ..., n − 1} do
3      for j ∈ {i + 1, ..., n − 1} do
4          if A[i] · A[j] = k then
5              c := c + 1
6          end
7      end
8  end
9  return c
```

```
LITHIUM(A : [ℕ; n]):
 1  for i ∈ {0, ..., n − 1} do
 2      if A[i] ≠ 0 then
 3          a : ℕ = A[i]
 4          A[i] := 0
 5          b : ℕ = LITHIUM(A)
 6          if a < b ∧ i < n − 1 then
 7              return b
 8          end
 9          return a
10      end
11  end
12  return 0
```

a) Gegeben sei das Array: $A = [6, 4, 25, 8, 9, 14, 10, 19, 8, 2]$. Führe jeden der drei Algorithmen auf A aus und gib den resultierenden Inhalt von A und (falls vorhanden) den Rückgabewert an. Verwende für HELIUM den Parameter $k = 42$. (3 Punkte)

b) Beschreibe jeden der drei Algorithmen in Worten. Bestimme und begründe außerdem deren asymptotische Laufzeiten. (5 Punkte)

Aufgabe 2 – Valleysort (6 Punkte)

Wir bezeichnen ein Array $A : [\mathbb{N}; n]$ als *valley-sortiert*, wenn für jedes Paar von Indizes $0 \leq i < j \leq n - 1$ folgendes gilt:

- wenn $j \leq \lfloor \frac{n-1}{2} \rfloor$, dann ist $A[i] \geq A[j]$
- wenn $i \geq \lceil \frac{n-1}{2} \rceil$, dann ist $A[i] \leq A[j]$

a) Füge die beiden valley-sortierten Arrays A_1 und A_2 zu einem valley-sortierten Array zusammen und gib das Resultat an. Entscheide und begründe, ob deine Lösung die einzige Lösung ist.

$$A_1 = [13, 8, 4, 5, 6] \text{ und } A_2 = [25, 13, 1, 1, 17, 26]$$

(2 Punkte)

b) Wir wollen nun Valleysort mithilfe von Mergesort aus der Vorlesung entwickeln. Unser Algorithmus soll statt `merge` nun `valleymerge` aufrufen, sonst aber identisch zu Mergesort sein und die gleiche asymptotische Laufzeit haben. Beschreibe die Funktion `valleymerge` und beweise dessen Korrektheit. (3 Punkte)

c) Warum hat Valleysort aus Teilaufgabe b) die gleiche asymptotische Laufzeit wie Mergesort? (1 Punkt)

Aufgabe 3 – Schnell Auswählen (6 Punkte)

Gegeben sei ein unsortiertes Array A, welches n natürliche Zahlen hält. Wir wollen das k-größte Element in A bestimmen.

a) Beschreibe einen Algorithmus, der für ein gegebenes Element $A[i]$ mit $i \in \{0, ..., n-1\}$ den Index bestimmt, an dem $A[i]$ stehen würde, wenn A absteigend sortiert wäre. Nenne und begründe die Laufzeit deines Algorithmus. (1 Punkt)

b) Entscheide und begründe, welche Werte die Konstante $b \in \mathbb{R}_+$ annehmen kann, sodass für die folgende Rekurrenz

$$T(n) = \begin{cases} \Theta(1), \text{ falls } n = 1 \\ T\left(\frac{n}{b}\right) + \Theta(n), \text{ sonst} \end{cases}$$

gilt, dass $T(n) \in O(n)$. (1 Punkt)

c) Beschreibe einen Algorithmus, welcher in $O(n)$ das k-größte Element in A bestimmt. Dein Algorithmus darf dabei die Reihenfolge der Einträge in A vertauschen. Du darfst annehmen, dass du den Median eines unsortierten Arrays in $\Theta(n)$ bestimmen kannst. Begründe, warum dein Algorithmus die geforderte Laufzeit hat. (4 Punkte)