Lehrstuhl für Informatik I
Algorithmen und Komplexität
Universität Würzburg

Würzburg, den 24. Oktober 2024

Prof. Dr. Alexander Wolff
Samuel Wolf, M. Sc.

2. Übungsblatt zur Vorlesung Algorithmen und Datenstrukturen (Winter 2024/25)

Aufgabe 1 – Aussagen zur O-Notation

Nehmen Sie kurz zu den folgenden Aussagen über einen fiktiven Algorithmus Stellung. Welche Aussagen sind möglicherweise sinnvoll und welche Aussagen sind in sich widersprüchlich oder nichtssagend? Begründen Sie Ihre Antwort. **4 Punkte**

a) Der Algorithmus hat im besten Fall eine Laufzeit von $\Theta(n)$ und im schlechtesten Fall eine Laufzeit von $\Theta(n^2)$.

b) Der Algorithmus hat im besten Fall eine Laufzeit von $\Omega(n)$ und im schlechtesten Fall eine Laufzeit von $O(n^2)$.

c) Der Algorithmus hat im besten Fall eine Laufzeit von $\Omega(n^2)$ und im schlechtesten Fall eine Laufzeit von $O(n)$.

d) Die Laufzeit des Algorithmus ist $O(n)$ und $\Omega(n^2)$.

e) Die Laufzeit des Algorithmus ist $\Omega(n)$ und $O(n^2)$.

f) Die Laufzeit des Algorithmus ist entweder $O(n^2)$ oder höher.

g) Der Algorithmus sortiert Folgen von bis zu 100 Zahlen mit einer Laufzeit von $\Theta(n \log n)$.

h) Die Laufzeit des Algorithmus ist $o(n)$ und $\omega(n)$.

Aufgabe 2 – O-Notation

Sei $f\colon \mathbb{N} \to \mathbb{R}$ eine Funktion. Beweisen oder widerlegen Sie folgende Behauptungen. Arbeiten Sie mit der Definition aus der Vorlesung, nicht mit Grenzwertbetrachtungen.

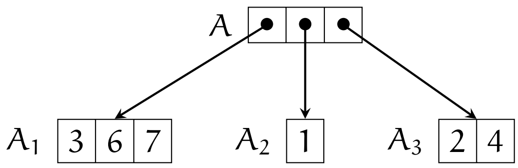
a) Für $f\colon n \mapsto \log_2(n^2)$ gilt $f \in \Theta(\log_{10} n)$. **3 Punkte**

b) Für $f\colon n \mapsto n \log n$ gilt $f \in \Omega(n^{1{,}1})$. **3 Punkte**

Aufgabe 3 – Merge mit mehreren Feldern

Seien $A_1, A_2, \dots, A_k$ Felder vom Typ ganze Zahl, wobei jedes der Felder aufsteigend sortiert ist und jedes Feld mindestens ein Element enthält. Sei weiterhin $n = A_1.\mathtt{length} + \dots + A_k.\mathtt{length}$. Gesucht ist ein Algorithmus, der angewandt auf $A_1, A_2, \dots, A_k$ ein aufsteigend sortiertes Feld B zurückgibt, das genau die Elemente aus $A_1, A_2, \dots, A_k$ enthält.

In dieser Aufgabe organisieren wir die Felder $A_1, \dots, A_k$ als *zweidimensionales* Feld A. Ein zweidimensionales Feld ist ein Feld, das Zeiger auf weitere Felder enthält. Zum Beispiel enthält das folgende Feld A drei Zeiger, die jeweils auf die Felder $A_1 = [3, 6, 7]$, $A_2 = [1]$ und $A_3 = [2, 4]$ zeigen.



Im Pseudocode schreiben wir int[ ][ ] A, um A als zweidimensionales Feld zu deklarieren. Analog zu gewöhnlichen Feldern können wir auf das i-te Feld aus A über $A[i]$ zugreifen. Dementsprechend erhalten wir das j-te Element aus dem i-ten Feld mittels $A[i][j]$. Im Beispiel gilt demnach $A[3][1] = 2$. Außerdem ist $A.\textit{length} = 3$.

a) Ihr Kommilitone schlägt folgenden Algorithmus vor:

```
int[ ] k-MergeIterativ(int[ ][ ] A)
    k = A.length
    n = A[1].length + ··· + A[k].length    // Eigentlich mit for-Schleife...
    B = new int[n]
    ℓ = 0
    for i = 1 to k do
        r = ℓ + A[i].length
        B[ℓ + 1 ... r] = A[i][1 ... A[i].length]    // Eigentlich mit for-Schleife...
        ℓ = r
    m = A[1].length
    for i = 2 to k do
        r = m + A[i].length
        Merge(B, 1, m, r)
        m = r
    return B
```

Wenden Sie den Algorithmus Ihres Kommilitonen auf das oben genannte Beispiel an. Geben Sie dabei die Indizes m und r sowie das Feld B nach jeder Iteration der zweiten Schleife an. **2 Punkte**

b) Analysieren Sie die Worst-Case-Laufzeit des Algorithmus aus Teilaufgabe a) in Abhängigkeit von n und k. Zeigen Sie, dass Ihre Abschätzung scharf ist, indem Sie Instanzen angeben, bei denen die Laufzeit des Algorithmus Ihrer Abschätzung entspricht. Begründen Sie, wieso Ihre Instanzen zu der angegebenen Worst-Case-Laufzeit führen. **4 Punkte**

c) Geben Sie einen Divide&Conquer-Algorithmus in kommentierten Pseudocode an, der dieses Problem in $O(n \log k)$ Zeit löst. Begründen Sie, wieso Ihr Algorithmus diese Laufzeit besitzt.

   Verwenden Sie folgende Funktionssignatur:

   $$\text{int[ ] } k\text{-}\textsf{MergeRekursiv}(\text{int[ ][ ] } A, \ell = 1, r = A.length)$$

   Hinweis: Sie dürfen davon ausgehen, dass k eine Zweierpotenz ist. Sie können den Merge-Algorithmus aus der Vorlesung als Unterprogramm benutzen. **4 Punkte**

Bitte geben Sie Ihre Lösungen bis **Donnerstag, 31. Oktober 2024, 14:00 Uhr** einmal pro Gruppe über Wuecampus als pdf-Datei ab. Vermerken Sie dabei stets die Namen und Übungsgruppen aller BearbeiterInnen auf der Abgabe.

Grundsätzlich sind stets alle Ihrer Aussagen zu begründen und Ihr Pseudocode ist stets zu kommentieren.

Die Lösungen zu den mit PABS gekennzeichneten Aufgaben, geben Sie bitte nur über das PABS-System ab. Vermerken Sie auf Ihrem Übungsblatt, in welchem Repository (*sXXXXXX*-Nummer) die Abgabe zu finden ist. Geben Sie Ihre Namen hier als Kommentare in den Quelltextdateien an.