

Präsenzübung 7

Besprechung: 24.11.25 – 28.11.25

Auf diesem Präsenzblatt wiederholen wir den bisherigen Stoff, als Vorbereitung auf die Probeklausur.

Aufgabe 1: Rekurrenzrelationen und Datenstrukturen (mündlich, keine Punkte)

Betrachten Sie einen Algorithmus, der ein Array der Größe n verarbeitet, indem er es in k gleich große Teile zerlegt, jeden Teil rekursiv bearbeitet und anschließend die Ergebnisse in Zeit $\Theta(n \log k)$ kombiniert.

- a) Stellen Sie die Rekurrenzgleichung für die Laufzeit $T(n)$ auf und lösen Sie diese mit dem Master Theorem für $k = 3$.
- b) Der Algorithmus speichert Zwischenergebnisse in einem binären Max-Heap. Nach der Verarbeitung enthält der Heap die Elemente

8	5	4	3	7	2	1
---	---	---	---	---	---	---

. Fügen Sie das Element 6 ein und führen Sie anschließend EXTRACTMAX aus. Geben Sie den Heap nach jeder Operation an.

Aufgabe 2: Graphalgorithmen (mündlich, keine Punkte)

Gegeben sei ein gerichteter Graph $G = (V, E)$ mit Kantengewichten $w : E \rightarrow \mathbb{R}_{>0}$. Sie möchten alle Knoten identifizieren, die von einem Startknoten s aus über Pfade erreichbar sind, deren sämtliche Kanten Gewichte im Bereich $[a, b]$ haben (mit $0 < a < b$). Entwickeln Sie einen effizienten Algorithmus, der diese Knoten findet. Geben Sie Pseudocode an und analysieren Sie die Laufzeit.

Aufgabe 3: Datenstruktur-Design (mündlich, keine Punkte)

Sie sollen eine Datenstruktur **PriorityDeque** entwerfen, die sowohl als Priority Queue als auch als Deque (double-ended queue) funktioniert. Die Datenstruktur soll folgende Operationen unterstützen:

- INSERTFRONT(x), INSERTBACK(x) - Element vorne/hinten einfügen
 - DELETEFRONT(), DELETEBACK() - Element vorne/hinten entfernen
 - EXTRACTMIN() - kleinstes Element entfernen und zurückgeben
 - GETMIN() - kleinstes Element zurückgeben (ohne zu entfernen)
1. Was sind die Laufzeiten der verschiedenen Operationen, wenn Sie die Datenstruktur mit einem Array oder einer einfach / doppelt verketteten Liste implementieren?
 2. Fällt Ihnen eine andere Implementierung ein, die effizienter ist? Welche grundlegenden Datenstrukturen würden Sie kombinieren? Begründen Sie Ihre Wahl. Beschreiben Sie die Implementierung Ihrer Datenstruktur und analysieren Sie die Laufzeiten aller Operationen.

Aufgabe 4: Bäume und Traversierungen (mündlich, keine Punkte)

Ein *gewichteter* k -närer Baum ist ein k -närer Baum, bei dem jeder Knoten zusätzlich ein Gewicht $w(v) \in \mathbb{R}_{>0}$ hat.

- a) Entwickeln Sie einen Divide-and-Conquer-Algorithmus, der für einen gegebenen vollständigen binären Baum der Höhe h den Pfad von der Wurzel zu einem Blatt findet, dessen Summe der Knotengewichte maximal ist. Analysieren Sie die Laufzeit.

- b) Sie erhalten einen ternären Baum in Level-Order:

5	3	7	2	1	4	6	8	9
---	---	---	---	---	---	---	---	---

. Die Zahlen sind die Knotengewichte. Definieren Sie eine neue Traversierungsreihenfolge, bei der für jeden Knoten gilt: Das mittlere Kind wird besucht, dann rekursiv alle Kinder mit kleinerem Gewicht (von links nach rechts), dann der Knoten selbst, dann alle Kinder mit größerem Gewicht. Was ist die Ausgabe?