

Übungsblatt 14

Hinweis. Ebenso wenig wie es möglich ist, all diese Fragen in einer Prüfung zu besprechen, ist es unmöglich, alle Fragen in der Übung im Detail zu diskutieren. Beantworten Sie bitte die Fragen für sich und fragen Unklarheiten nach.

Aufgabe 1 (Landau-Symbole).

1. Definieren Sie für Funktionen f und g , was $f \in O(g)$, $f \in o(g)$, $f \in \Omega(g)$ und $f \in \omega(g)$ bedeutet.
2. Drücken Sie die folgenden Aussagen mittels Landau-Symbole aus:
 - (a) Die Funktion $f(n)$ wächst schneller als jedes Polynom.
 - (b) Die Funktion $f(n)$ wächst schneller als jede lineare Funktion, aber langsamer als jede quadratische Funktion.
 - (c) Die Funktion $f(n)$ wächst nicht schneller als die Funktion $g(n) = 2^n$.

Aufgabe 2 (Divide-And-Conquer).

1. Nennen und beschreiben Sie einen Sortieralgorithmus, der nach dem *Divide-and-Conquer*-Prinzip funktioniert.
2. Wie ist die Worst-Case-Laufzeit Ihres genannten Algorithmus? Wie ist die Average-Case-Laufzeit?
3. Was ist die beste untere Schranke für den Worst-Case, die vergleichsbasierte Sortieralgorithmen erreichen können?
4. Was besagt das *Mastertheorem*? Wofür wird es genutzt?
5. Welche weiteren algorithmischen Probleme kennen Sie, die per Divide-And-Conquer-Strategie gelöst werden?

Aufgabe 3 (Greedy).

1. Was bedeutet es, wenn eine Lösungsstrategie *greedy* ist?
2. Unter welchen Umständen konstruiert ein solcher Ansatz mit Sicherheit eine optimale Lösung zum gegebenen Problem?

3. Was ist ein Matroid? Nennen Sie ein Beispiel.
4. Beschreiben Sie Kruskals Algorithmus.
5. Beschreiben Sie Dijkstras Algorithmus.

Aufgabe 4 (Datenstrukturen).

1. Was ist ein *Fibonacci-Heap*? Wozu kann er verwendet werden?
2. Was ist eine *Union-Find*-Datenstruktur? Wozu kann sie verwendet werden?

Aufgabe 5 (Dynamische Programmierung).

1. Welches Paradigma beschreibt die *Dynamischen Programmierung*?
2. Nennen Sie ein Problem, das sich mittels dynamischer Programmierung lösen lässt.
3. Was ist ein *optimaler Suchbaum*, und wie lässt er sich konstruieren? Welche Laufzeit hat der Algorithmus?
4. Beschreiben Sie den Floyd-Algorithmus.

Aufgabe 6 (Parallele Algorithmen).

1. Was ist die Intuition hinter *Nick's Class*? Definieren Sie **NC**. Welche algorithmischen Probleme gehören zu **NC**?
2. Wie können Sie die Multiplikation zweier $n \times n$ -Matrizen effizient parallelisieren?
3. Gehört das Sortieren einer n -elementigen Liste zu **NC**?