
Algorithmen und Berechnungskomplexität I WS 15/16

Universität Bonn, Institut für Informatik, Abteilung I

2. Aufgabenblatt zur Vorlesung

Abgabe: 3.11. (12³⁰)

- Unter ***www.i1.informatik.uni-bonn.de*** werden Unterlagen zu Vorlesung und Übung bereitgestellt.
- Bearbeitung und Abgabe der Übungsblätter ist in festen Gruppen von bis zu 3 Personen erlaubt.
- Auf den Abgaben müssen die Namen aller Teilnehmer und die Nummer der Übungsgruppe (siehe Homepage) stehen.
Achtung: Abgaben ohne diese Informationen werden nicht berücksichtigt!
- Lösungen können am Tag der Abgabe bis zum Beginn der Vorlesung in den Postkasten im AVZ III eingeworfen werden.

Aufgabe 5: Wachstum von Funktionen (4 Punkte)

Geben Sie für die folgenden Funktionen f_i jeweils möglichst einfache Funktionen g_i an, so dass $f_i \in \Theta(g_i)$ gilt.

- $5n^2 - 13n^{2,5} \cdot \log n^{2,5} - 8n^4 + 0,01n^5$
- $\prod_{i=0}^{\infty} n^{(2^{-i})}$
- $f_3(n) := n \cdot (\sqrt{n} + \log_3 n)$
- $f_4(n) := a \cdot n^{k_1} + b \cdot n^{k_2}$ für Konstanten $a, b, k_1, k_2 > 0$

Aufgabe 6: Entwurf und Analyse eines Algorithmus(4 Punkte)

Gegeben seien n Punkte in der Ebene. Geben Sie einen Divide-and-Conquer-Algorithmus an, der die Ebene so lange in Flächen unterteilt, bis in jeder Unterteilungsfläche nur noch *genau* ein Punkt liegt. Eine vorhandene Region darf hierbei durch eine Gerade (die endet, sobald sie eine Regionsgrenze schneidet) in zwei Teile aufgeteilt werden. Dabei soll die Unterteilungshierarchie in einem Baum abgelegt werden, siehe Abbildung 1. Nehmen Sie dabei an, dass folgende Prozeduren existieren:

- *TeileFläche(Fläche f)*: Teilt eine Fläche f mit mehr als einem Punkt in zwei Teilflächen, die jeweils mindestens einen Punkt enthalten. Die Teilflächen werden als Tupel zurückgegeben.
- *GibPunkt(Fläche f)*: Falls eine Fläche f genau einen Punkt enthält wird dieser zurück gegeben. Andernfalls eine leere Referenz.

Wie groß ist die Tiefe des Baumes im Worst-Case?

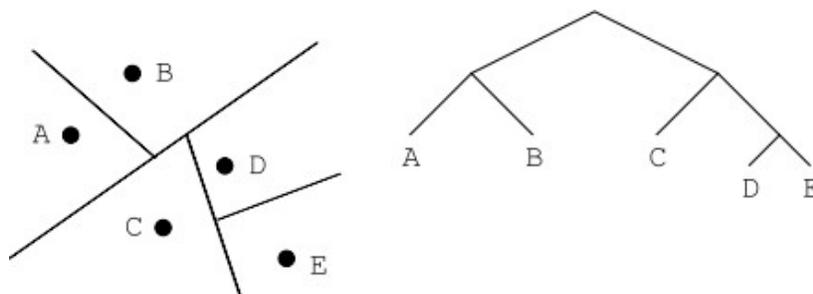


Abbildung 1: Beispiel zu Aufgabe 3.

Aufgabe 7: MergeSort (4 Punkte)

Sortieren Sie das folgende Array unter Verwendung des in der Vorlesung vorgestellten Merge-Sort-Algorithmus.

[54, 36, 42, 1, 52, 87, 77, 28, 101, 6]

Verwenden Sie zur Beschreibung der Divide- bzw. Merge-Schritte jeweils einen Binärbaum. Geben Sie außerdem die Anzahl benötigter Vergleiche pro Merge-Schritt und für alle Merge-Schritte insgesamt an.

Hinweis: Jeder Knoten enthält ein Array; Kanten verweisen auf die Ergebnisse des vorherigen Funktionsaufrufs. Im Wurzelknoten steht ein Array der Länge 10 (unsortiert bzw. sortiert).

Aufgabe 8: Dichtestes Punktepaar auf einer Geraden (4 Punkte)

Betrachten Sie den Divide- and Conquer Algorithmus zum Bestimmen eines dichtesten Punktepaars in der Ebene, wie er in der Vorlesung vorgestellt

wurde. Wandeln Sie diesen Algorithmus so ab, dass ein dichtestes Punktepaar von Punkten bestimmt wird, die sich alle auf einer horizontalen Geraden befinden. Beachten Sie hierbei folgende Aspekte:

- Welche Schritte des Algorithmus können vereinfacht oder weggelassen werden? Wie kann insbesondere der Merge-Schritt vereinfacht werden? Die Abstände welcher Elemente werden jeweils berechnet?
- Welche Laufzeit hat das neue Verfahren, und welcher Anteil ist dominierend?

Geben Sie anschließend einen Algorithmus an, der nicht auf dem Divide- and Conquer Prinzip beruht, und, basierend auf den vorherigen Überlegungen, eine intuitivere Herangehensweise an das Problem „Dichtestes Punktepaar auf einer Geraden“ umsetzt.