Algorithmen und Berechnungskomplexität I, WS 12/13
Aufgabenblatt 3
Universität Bonn, Institut für Informatik, Abteilung I

- *Die Lösungen können bis Dienstag, 6.11., 12:15 Uhr in den Postkasten im AVZ III eingeworfen werden (vom Haupteingang im kleinen Raum auf der linken Seite). Gebt bitte immer gut sichtbar auf dem Deckblatt die Gruppennummer (A-I) an, wie auf der Vorlesungswebseite angegeben.*

Aufgabe 9: **Untere Schranken** **(4 Punkte)**

a) Sei $f(n), n \in \mathbb{N}$, die Folge der Fibonacci-Zahlen, definiert als:

$$\begin{aligned} f(0) &= 1 \\ f(1) &= 1 \\ f(n) &= f(n-1) + f(n-2). \end{aligned}$$

Erfolge die Berechnung so, wie in Skript auf Seite 32 angegeben, also ohne Wiederverwendung von Teilergebnissen. Zeigen Sie: Für die Anzahl Knoten $K(n)$ des Rekursionsbaums zur Berechnung von $f(n)$ gilt: $K(n) \geq \frac{1}{2} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n$.

b) Wie im Skript auf Seite 35 ersichtlich, ist die Matrizenpultiplikation assoziativ – mit anderen Worten ist die Klammerung für das Ergebnis irrelevant. Wohl aber ist die Klammerung für den Aufwand der Berechnung des Multiplikationsergebnisses relevant: Es gilt, eine Klammerung zu finden, die möglichst wenig Aufwand verursacht. Zeigen Sie, dass die im Skript angegebene Rekursionsgleichung

$$M(n) := \sum_{k=1}^{n-1} M(k) \cdot M(n-k),$$

mit $M(1) = 1$, in $\Omega(2^n)$ liegt.

Bitte wenden!

Aufgabe 10: Master-Theorem (4 Punkte)

Verwenden Sie das Master-Theorem, um für die folgenden Rekursionsgleichungen asymptotische Schranken anzugeben:

a) $T(n) = 2T(n/4) + 1$.

b) $T(n) = 2T(n/4) + \sqrt{n}$.

c) $T(n) = 2T(n/4) + n$.

d) $T(n) = 2T(n/4) + n^2$.

Aufgabe 11: Dichtestes Punktepaar auf einer Geraden (4 Punkte)

Betrachten Sie den Divide- and Conquer Algorithmus zum Bestimmen eines dichtesten Punktepaares 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.

Aufgabe 12: Dynamische Programmierung (4 Punkte)

Gegeben sei eine Familie M von n Münzen mit Werten $v_1, v_2, ..., v_n \in \mathbb{N}$. Sie sollen nun einen bestimmten Betrag S mit Hilfe dieser Münzen zusammensetzen, d.h eine Teilmenge $U \subseteq 1, 2, ..., n$ finden, so dass $S = \sum_{i \in U} v_i$. Jede Münze darf dabei nur einmal verwendet werden.

Geben Sie ein Verfahren an, das mittels Dynamischer Programmierung die minimale Anzahl an Münzen aus M bestimmt, die dazu nötig ist.