TU Ilmenau, Fakultät IA
FG Telematik/Rechnernetze
Prof. Dr.-Ing. G. Schäfer
https://www.tu-ilmenau.de/telematik/pua

TECHNISCHE UNIVERSITÄT
ILMENAU

Programmierung und Algorithmen WS 25/26

Übungsblatt 10

Die Lösungen der Aufgaben sind bis zum 18.01.26, 23:59 Uhr abzugeben.

Die Besprechung der Aufgaben erfolgt in KW 4.

Aufgabe 1 (Divide-and-Conquer-Algorithmen) **4 + 3 Punkte**

Ein intuitiver iterativer Algorithmus für die Berechnung der Potenz $a^k, k \in \mathbb{N}_0$ benötigt $\Theta(k)$ Multiplikationsoperationen.

(a) Entwerfen Sie einen *rekursiven Divide-and-Conquer*-Algorithmus in Java, welcher die Potenz a^k mit $\mathcal{O}(\log k)$ Multiplikationsoperationen berechnet.

(b) Geben Sie eine Rekurrenzrelation für die maximale Anzahl an Multiplikationsoperationen Ihres Algorithmus' an und beweisen Sie, dass Ihr Algorithmus die geforderte Schranke einhält.

Hinweis: Divide-and-Conquer-Algorithmen unterteilen das zu lösende Problem in kleinere Teilprobleme, lösen diese separat und konstruieren aus den Teillösungen eine Lösung für das ursprüngliche Problem. Für die Lösung der Teilprobleme kommt in der Regel Rekursion zum Einsatz.

Aufgabe 2 (Suchen von Paaren #2) **4 + 3 Punkte**

Im vorigen Übungsblatt sollte in Aufgabe 5 ein Algorithmus zum Finden von Zahlenpaaren hergeleitet werden. Für diese Aufgabe soll ein verwandtes Problem betrachtet werden.

Gegeben sei ein *schon sortiertes* Array A der Länge n aus ganzen Zahlen ohne Duplikate, sowie eine beliebige Zahl $z \in \mathbb{Z}$. Gesucht ist die Menge aller Zahlenpaare (x, y) mit $x, y \in A$, sodass $x + y \leq z$ und kein Zahlenpaar $x', y' \in A$ existiert, für welches $x' + y' > x + y$ gilt. Um Duplikate zu vermeiden, soll außerdem $x \leq y$ gelten. Wenn kein Zahlenpaar diese Anforderungen erfüllt, ist die Lösung für das Problem $\emptyset$.

Beispielsweise ist die Lösung für $A = (1, 3, 5, 10)$ und $z = 7$ die Menge $\{(1, 5), (3, 3)\}$.

(a) Beweisen Sie durch vollständige Induktion, dass das beschriebene Problem gelöst werden kann.

(b) Aus dem Induktionsbeweis kann ein Algorithmus hergeleitet werden. Geben Sie eine geeignete Rekurrenzrelation an, um die Anzahl der benötigten Additionen eines so hergeleiteten Algorithmus abzuschätzen. Begründen Sie kurz.

Hinweis: Ein möglichst effizienter Algorithmus sollte die gegebenen Eigenschaften des Arrays (Sortiertheit, Duplikatfreiheit) ausnutzen.

Hinweis 2: Eine geschlossene Form zur Abschätzung ist nicht gefordert.

Bitte wenden!

Aufgabe 3 (Induktiver Algorithmenentwurf) **1 + 2 + 5 Punkte**

Gegeben Sei eine Folge $a = x_0, ..., x_{n-1}$ bestehend aus n natürlichen Zahlen, wobei einzelne Zahlen mehrmals vorkommen können. Gesucht ist ein Element x_i aus a, welches insgesamt echt mehr als $n/2$ mal in der Folge vorkommt (d.h. es wird ein Element mit *absoluter* Mehrheit gesucht, im Folgenden auch kurz *Mehrheitselement* genannt), beziehungsweise die Aussage, dass kein solches Element in a existiert. Die einzige erlaubte Vergleichsoperation *für die Elemente der Folge* sei zudem ein paarweiser Vergleich der Art „ist x ein Duplikat von y?“ (also $x = y$?).

Hinweis: Sonstige Vergleiche, wie zum Beispiel der Vergleich von Zählvariablen für eine Implementierung, sind von der letzten Einschränkung nicht betroffen!

Gegeben sei ferner folgender *konstruktiver* Induktionsbeweis, dass das Problem für alle Folgen der Länge $n = 2^k, k \geq 0$ lösbar ist (die Länge ist also eine Zweierpotenz).

IA: $n = 1$, d.h. $k = 0$
$\Rightarrow$ Die Folge besteht also nur aus dem Element x_0, welches auch ein Mehrheitselement ist.

IV: Wir wissen wie das Problem für ein beliebiges $n = 2^k, k \geq 0$ gelöst werden kann.

IBeh: Das Problem kann auch für Folgen der Länge $2n = 2^{k+1}$ gelöst werden.

IS: $n \to 2n$, d.h. $k \to k + 1$
Wir teilen die Folge a der Länge $2n = 2^{k+1}$ zunächst in zwei gleich große Folgen a_l und a_r auf, jeweils mit der Länge $2\frac{n}{2} = n = 2^k$ (zum Beispiel in der Mitte). Gibt es in a ein Mehrheitselement, dann muss dieses auch in a_l oder in a_r ein Mehrheitselement sein (sonst wäre seine Gesamtanzahl in a kleiner oder gleich $\frac{n}{2} + \frac{n}{2} = n$, im Widerspruch zur Annahme, dass es ein Mehrheitselement ist, also mehr als n mal vorkommt.)
Demnach bestimmen wir zunächst mit Hilfe der IV für die beiden Teilfolgen a_l und a_r der Länge $n = 2^k$, ob es ein Mehrheitselement gibt oder nicht (und falls ja, auch den Wert des Mehrheitselements). Danach haben wir entweder 0, 1 oder 2 Elemente, welche als Mehrheitselement im Gesamtfeld a in Frage kommen. Zuletzt *zählen* wir die Gesamtanzahl jedes dieser möglichen Kandidaten in a, wofür paarweise Vergleiche der Form „$x = y$?“ genügen. Kommt einer der Kandidaten dabei mehr als $2\frac{n}{2} = n$ mal vor, so ist dies ein Mehrheitselement in a, ansonsten gibt es in a kein Mehrheitselement.

□

Lösen Sie nun folgende Aufgaben:

(a) Geben Sie die Rekurrenzrelation für die Anzahl an paarweisen Vergleichen von *Elementen der Folge* des *rekursiven* Algorithmus an, welcher der Form des Induktionsbeweises folgt. Gehen Sie dabei vom *worst-case* aus.

(b) Welche *asymptotische* Laufzeit ergibt sich somit für den resultierenden Algorithmus? Begründen Sie Ihre Antwort, sonst kann sie nicht gewertet werden!

(c) Implementieren Sie den resultierenden rekursiven Algorithmus in Java. Schreiben Sie dazu eine Methode `public static int majority(int[] a)`, welche ein Mehrheitselement in `a` berechnet und zurückgibt, bzw. feststellt, dass kein Mehrheitselement existiert. Im letzteren Fall soll der Rückgabewert der Methode `-1` sein.

Wie im Induktionsbeweis dürfen Sie davon ausgehen, dass `a.length` $= 2^k, k \geq 0$ gilt.

Hinweis: Sie werden eine rekursive Hilfsmethode benötigen.

Aufgabe 4 (Induktiver Algorithmenentwurf) **4 + 2 Punkte**

Eine Fläche der Größe $2^k \times 2^k, k \geq 0$ soll vollständig gefliest werden. Für diese Aufgabe stehen Ihnen eine Einzelfliese (1×1) und beliebig viele L-förmige Fliesen mit einer Fläche von 3 und einer Kantenlänge von 2 (siehe Abbildung 1) zur Verfügung. Ihr Chef (ein erfahrener Fliesenleger) besteht darauf, dass dieses Problem immer lösbar sei. Können Sie ein allgemeines Verfahren finden?

(a) Beweisen Sie *konstruktiv* durch vollständige Induktion, dass das Fliesenproblem immer lösbar ist.

(b) Zeigen Sie, dass genau $\frac{1}{3}\left(2^{2k} - 1\right)$ L-förmige Fliesen benötigt werden.

Abbildung 1: Beispiellösung für $k = 2$ mit Einzelfliese und 5 L-förmigen Fliesen (Schattierungen haben keine weitere Bedeutung und dienen nur der Erkennbarkeit individueller Fliesen).

Hinweis: Hier bietet sich die *Divide-and-Conquer*-Idee an. Möglicherweise ist es sinnvoll mehr als 2 Teilprobleme zu erzeugen.