

Algorithmen und Datenstrukturen Klausur — 2. August 2019 (Gedächtnisprotokoll)

(Es sind ca. 80% von den Aufgaben aus Klausur)

Aufgabe 1: ADT (10 Punkte)

ADT Circle (als 3-Tupel) mit Radius, X-Koordinate, Y-Koordinate (ganzzahlig)

1. Formale Spezifikation von ADT Circle angeben mit vorgegebenen informalen Funktionen:
 - new: neuen Circle erstellen mit Radius 1 und Koordinaten (0,0)
 - move: neue Koordinaten zuweisen
 - scale: Radius ändern
 - area: Gibt Flächeninhalt zurück
 - touches: Überprüfung ob sich zwei Kreise berühren (Rückgabewert: boolean)
2. Geben Sie die Interpretation (als mathematische Funktion) für area und move an

Aufgabe 2: QuickSort (5 Punkte)

1. Es ist eine Liste (Sequenz) gegeben bestehend aus ganzen Zahlen. Schreiben sie die Schritte für QuickSort auf, entweder in-place nach Vorlesung oder vereinfachter Version aus der Übung. Alle Teilsequenzen!

Aufgabe 3: Rekursionsgleichung

$$t(n) = 8t\left(\frac{n}{2}\right) + n^2$$

$$t(1) = 3$$

1. Rekursionsgleichung lösen. Liegt $t(n)$ in $O(n^3)$?
2. $f(n) \in O(g(n))$

$$f(n) = n^2 * \log n * \log n$$

$$g(n) = n^3$$

Beweisen sie die Gültigkeit

(Tipp: Ableitung von $(\log n)^2$ ist $(2 \log n) / n$)

Aufgabe 4: NP-vollständig

Gegeben seien Mengen M , N und $A \subseteq M$, $B \subseteq N$ und zwei Probleme:

Q:

Eingabe: $x \in M$

Ausgabe: 1 falls $x \in A$, 0 sonst.

R:

Eingabe: $x \in N$

Ausgabe: 1 falls $x \in B$, 0 sonst.

Was ist für sie eine Transformation?

1. Wann ist ein Problem NP-vollständig?
2. Sind NP-vollständig die schwierigsten oder die einfachsten Probleme in NP?
3. Kantenüberdeckung Algorithmus angeben für NP
4. Gibt es einen Algorithmus für $P = NP$

Aufgabe 5: Laufzeit / Zeitkomplexität

1. Knotenreihenfolge angeben für Preorder, Levelorder, Inorder eines Binärbaums
2. Unter welcher Bedingung funktioniert binäre Suche?
3. Laufzeit von Worst Case von Binärer Suche angeben
4. Wieviel Blätter hat ein Baum höchstens und mindestens der Tiefe 6?
5. Wieviel Knoten hat ein Baum höchstens und mindestens der Tiefe 6?
6. Welche Tiefe hat ein Baum mit 5 Blättern höchstens und mindestens?
7. Laufzeit Worst Case eines AVL Baums mit Knotenanzahl n
8. Laufzeit Worst Case eines binären Suchbaums mit Knotenanzahl n

Aufgabe 6 Greedy & Dynamische Programmierung

1. Matroid Austauscheseigenschaft angeben
2. Was ist der Vorteil von einem Matroid
3. Rekursionsgleichung gegeben

$$f(0,y) = 1 \text{ wenn } y \geq 0$$

$$f(x,0) = 0 \text{ wenn } x > 0$$

$$f(x,y) = f(x-1,y) + f(x,y-1) \text{ sonst}$$

- Wie oft wird $f(1, 1)$ beim Aufruf von $f(2, 2)$ aufgerufen bei Rekursion
 - Tabelle für Dynamische Programmierung erstellen für $f(3,3)$
 - Wie oft wird bei Dynamischer Programmierung $f(1,1)$ bei $f(2, 2)$ aufgerufen
4. Python Code für Summe von Strings in einer Liste (mit Teile & Herrsche)
- Laufzeit angeben im Worst case

Aufgabe 7 Einfach verkettete Liste (10 Punkte)

1. Geben Sie in Pseudocode die Funktion `addList(li, elem)` an, die bei einer einfach verketteten Liste ein neues Listenelement erstellt und es an das letzte Element der Liste anhängt. Die einfach verkettete Liste besteht aus `(value, next)` Tupeln wobei `value` den Wert speichert und `next` der Zeiger auf das nächste Listenelement.

Verfügbare Funktionen:

- `content(p)` : Dereferenzieren vom Zeiger `p`
- `head`: Gibt den Zeiger auf das erste Listenelement von der Liste zurück
- `getValue(elem)`
- `setValue(elem, value)`
- `getNext(elem)`
- `setNext(elem, next)`
- `address(n)`

Aufgabe 8: Bäume

1. Definieren Sie minimaler Spannbaum