Ludwig-Maximilians-Universität München
Institut für Informatik
Prof. Dr. Thomas Seidl
Janina Sontheim, Florian Richter

München, 07.05.2019

Algorithmen und Datenstrukturen

SS 2019

Übungsblatt 3: Grundlagen

Aufgabe 3-1 *Mastertheorem*

Nutzen Sie das Mastertheorems um eine Laufzeitabschätzung zu geben. Sollte dies nicht möglich sein, begründen Sie warum.

(a) $T(n) = 8T(n/2) + n * log(n)$

(b) $T(n) = 7T(n/3) + n^2 + 4n + 1$

(c) $T(n) = n * T(n/2) + 1$

Aufgabe 3-2 *Laufzeit*

Gegeben sei folgender Algorithmus, der das Maximum Subarray Problem löst:

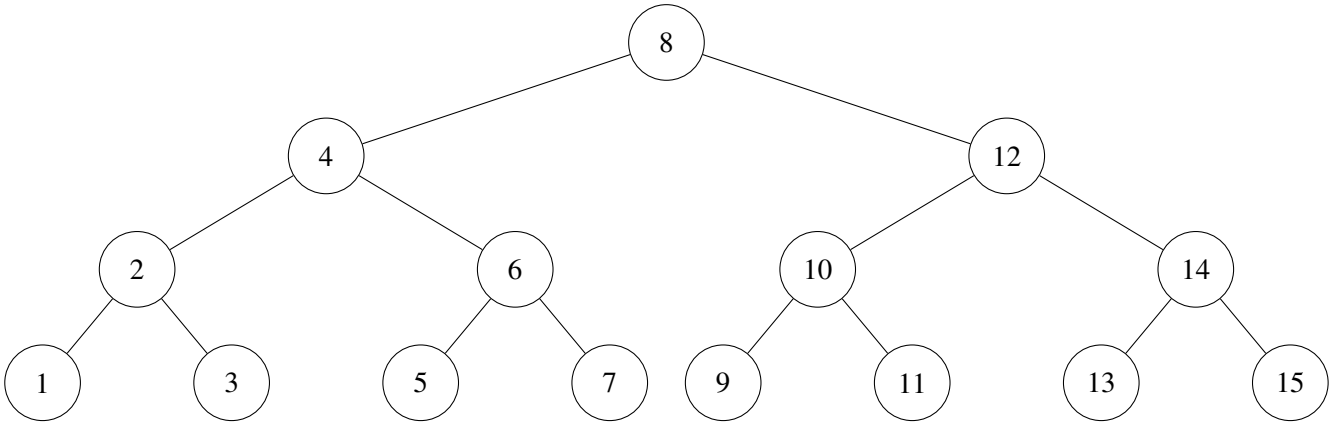
```
public static int msp(int[] array) {
        int result = 0;
        int n = 0;

        for (int i = 0; i < array.length; i++){
            n = 0;
            for (int j = i; j < array.length; j++){
                n += array[j];
                if(n > result){
                    result = n;
                }
            }
        }
        return result;
}
```

(a) Welche worst-case Laufzeitkomplexität hat dieser Algorithmus?

(b) **Bonus:** Überlegen Sie sich, ob sie den Algorithmus auch effizienter gestalten können.

Aufgabe 3-3 *Eigenschaften von Bäumen*

Gegeben ist folgender Baum:



Hinweis: Die Höhe h wird wie in der Vorlesung definiert!
Beantworten Sie nun folgende Fragen. Achten Sie wo nötig auf eine formal korrekte Herleitung!

(a) Um welche Art von Baum handelt es sich? Welche Arität (=Stelligkeit) besitzt er?

(b) Welche Höhe h hat dieser Baum? Wie viele Knoten k und wie viele Blätter b besitzt er?

(c) Angenommen man würde die Höhe um 2 erhöhen, wie viele Knoten k muss der Baum nun mindestens haben und wie viele kann er maximal haben, wenn sich die bestehenden Einträge des Baumes dabei nicht ändern?

(d) Wie viele Knoten k hat der Baum mindestens und maximal, wenn die Höhe nun um n erhöht wird?

(e) Wie ändern sich die Formeln aus der Vorlesung für:

(i) Die minimale Knotenanzahl k_{min}

(ii) Die maximale Knotenanzahl k_{max}

(iii) Die maximale Anzahl der inneren Knoten k_{in}

(iv) Die maximale Anzahl der Blätter b

wenn wir anstelle eines Baums, in denen jeder Knoten maximal 2 Nachfolger hat, einen Baum mit Arität n annehmen? Geben Sie die allgemeinen Formeln in Abhängigkeit der Höhe h an.

(f) **Bonus:** Die in der Vorlesung vorgestellten Baumtraversierungsmöglichkeiten haben alle lineare Laufzeit in Bezug auf die Knotenanzahl k des Baumes. Begründen Sie warum. Wie viele Schritte brauchen Sie im Durchschnitt, wenn Sie ein Element des Baumes auf diese Weise suchen wollen?

(g) **Bonus:** Angenommen alle Einträge sind wie im obigen Baum der Größe nach sortiert, wie könnte dann ein effizienter Algorithmus aussehen, um ein beliebiges Element des Baumes zu suchen? Welche Laufzeit hätte der Algorithmus? Und wie ändert sich die Laufzeit bei einem allgemeinen Baum mit Arität n?