

Algorithmen und Datenstrukturen, Übung 6

Marouane Soussi, Lars Happel, Mustafa Miresch

Mai 2022

Aufgabe 1

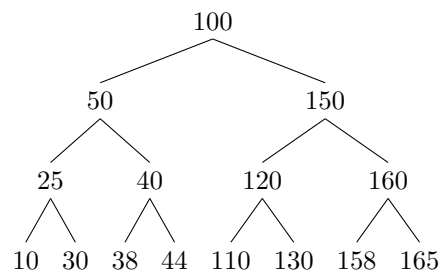
siehe .java Datei

Aufgabe 2

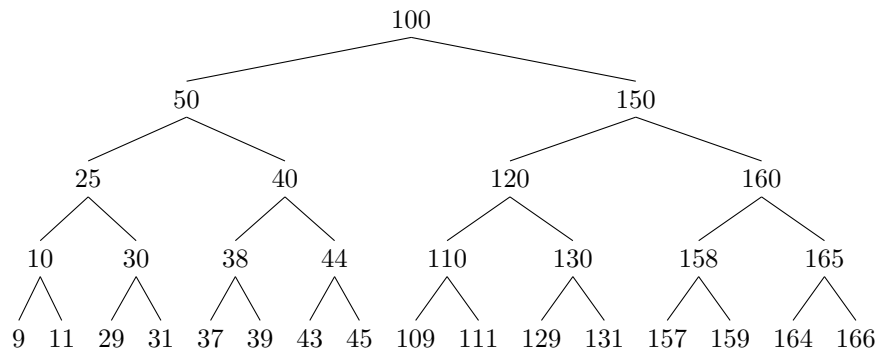
a)

1.

Baum vor der Frühlingsbaum-Operation (Höhe = 3, Blätter = 8):



Baum nach der Frühlingsbaum-Operation (Höhe = 4, Blätter = 16):



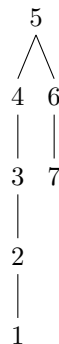
Bei 8 vorhandenen Blättern kamen also 16 neue Blätter hinzu.

2.

Baum vor der Operation (Höhe 3, Blätter 2):



Baum nach der Operation (Höhe 4, Blätter 2):



(Anmerkung: Die Knoten unter der 4 sind linke Kinder, Knoten unter der 6 ist ein rechtes Kind.)
Es wurden also nur 2 Blätter hinzugefügt, da aufgrund der paarweisen Verschiedenheit der Knoten 2 kein rechtes Kind mehr bekommen konnte, und der Knoten 6 kein linkes Kind bekommen konnte.

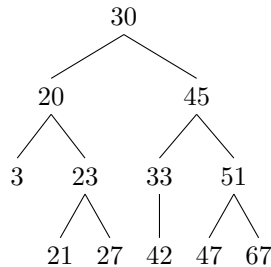
b)

Beschreibung: Der Algorithmus bekommt einen Knoten des Baumes als Eingabe. Er prüft, ob der Knoten ein linkes bzw. rechtes Kind hat und falls ja, ruft er sich rekursiv auf diesen auf. Falls der Knoten keine Kinder hat (also ein Blatt ist) dann holt der Algorithmus sich die Werte von Successor und Predecessor des aktuellen Knotens und vergleicht diese mit dem Wert des Knotens. Vorausgesetzt der Abstand zum Predecessor/Successor ist größer als 1, wird ein linkes bzw. rechtes Kind erstellt. Der Abstand muss geprüft werden, damit alle Knoten paarweise verschiedene Werte behalten.

Laufzeit: Die Laufzeit ist $O(n^2)$. Begründung: Die Anzahl der Blätter in einem Binary-Tree ist maximal $\frac{n}{2}$. Innerhalb jedes Blattes werden ausschließlich die Funktionen Predecessor und Successor verwendet, welche beide je $O(n)$ Laufzeit haben. Alle weiteren Zeilen/Operationen haben eine konstante Laufzeit von $O(1)$. Insgesamt ergibt sich also eine Laufzeit von $O(\frac{n}{2} * 2n + c) = O(n^2)$

Aufgabe 3

a)



Anmerkung: 42 ist das rechte Kind von 33.

b)

Uns ist bekannt, dass das letzte Element des Array die Wurzel des Gesamtbaums sein muss. Mit dieser Information lässt sich das Array in zwei Bereiche unterteilen:

Links (= linker Teilbaum) := Alle Elemente die kleiner als die Wurzel sind
und

Rechts (= rechter Teilbaum) := Alle Elemente die größer als die Wurzel sind.

Die so gefunden Subarrays Links und Rechts können selbst wieder als Baum behandelt werden. Ihre jeweiligen rechtesten Einträge stellen die Wurzel der Teilbäume dar (und somit die Kinder der eben gefunden Wurzel der höheren Ebene).

Mittels des oben beschriebenen Teilungsverfahrens lassen sich rekursiv alle linken bzw. rechten Kinder ermitteln. So kann der Baum eindeutig aus der Postorder Notation wiederhergestellt werden.