

Effiziente Algorithmen / Theoretische Informatik III

5. Übung

1. Aufgabe:

Wir betrachten einen Rot-Schwarz-Baum mit Tiefe $2n$.

- (a) Wie viele Knoten hat der Baum höchstens?
- (b) Wie viele Knoten hat der Baum mindestens?

2. Aufgabe:

Wir beginnen mit einem leeren Rot-Schwarz-Baum.

Wie sieht der Rot-Schwarz-Baum nach Einfügen der Elemente 41, 38, 31, 12, 19, 8 aus?

3. Aufgabe:

Wir beginnen mit einem leeren Rot-Schwarz-Baum und fügen $n > 1$ Elemente ein.

Zeigen Sie, dass der entstehende Rot-Schwarz-Baum mindestens einen roten Knoten hat.

4. Aufgabe:

Seien Elemente $e_1 < e_2 < \dots < e_n$ gegeben. Sei außerdem eine Folge von **Finde**-Operationen gegeben, in der jedes Element e_i insgesamt H_i Mal gefunden werden muss. Ein *optimaler statischer binärer Suchbaum* ist ein binärer Suchbaum, in dem die Gesamtzahl der Zugriffe auf die Elemente im Baum für die gegebene Liste an **Finde**-Operationen minimal ist.

Zum Beispiel ist der folgende Suchbaum optimal, wenn die Elemente 4 und 7 zwei Mal gefunden werden müssen und die Elemente 9, 12 und 17 je ein Mal gefunden werden müssen:

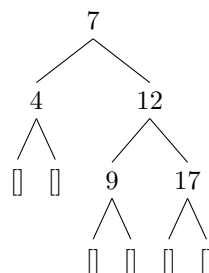


Abbildung 1: Beispiel Suchbaum

Für jede der 7 **Finde**-Operationen muss die Wurzel aufgerufen werden. Die 4 wird zwei Mal aufgerufen, die 12 drei Mal (ein Mal für jedes Element größer als 7), und 9 und 17 werden jeweils einmal aufgerufen.

Finden Sie alle möglichen Suchbäume für die drei Elemente 1, 2, 3, die optimal sein können. Finden Sie zu jedem dieser Suchbäume eine entsprechende Häufigkeitsverteilung.