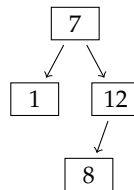


7. Aufgabe: Heap und binärer Suchbaum

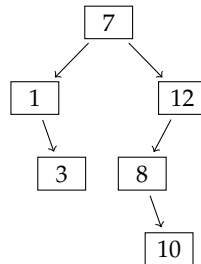
(a)

- (i) Fügen Sie nacheinander die Zahlen 7, 1, 12, 8, 10, 3, 5 in einen leeren binären Suchbaum ein und zeichnen Sie den Suchbaum nach „8“ und nach „3“.

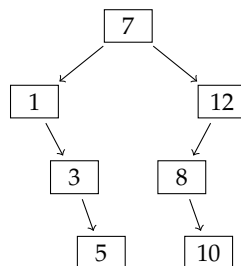
Nach Einfügen von „8“:



Nach Einfügen von „3“:

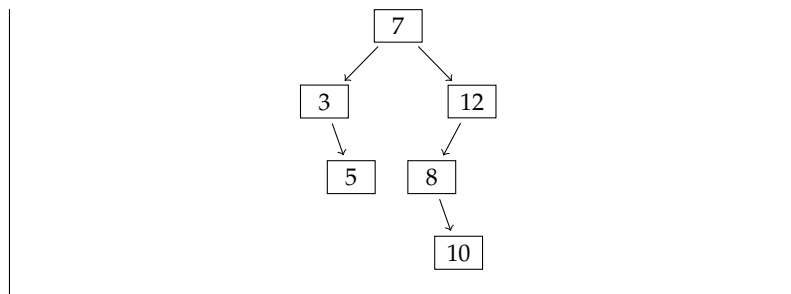


Nach Einfügen von „5“:



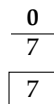
- (ii) Löschen Sie die „1“ aus dem in (i) erstellten Suchbaum und zeichnen Sie den Suchbaum.

Nach Löschen von „1“:

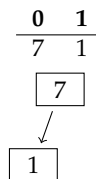


- (iii) Fügen Sie 7, 1, 12, 8, 10, 3, 5 in einen leeren MIN-Heap ein, der bzgl. „ \leq “ angeordnet ist. Geben Sie den Heap nach jedem Element an.

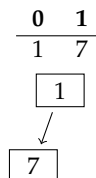
Nach dem Einfügen von „7“:



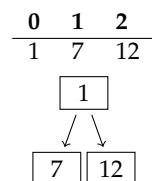
Nach dem Einfügen von „1“:



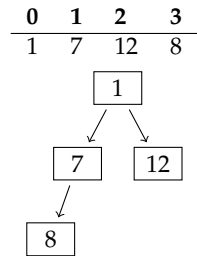
Nach Vertauschen von „1“ und „7“:



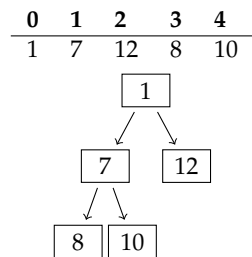
Nach dem Einfügen von „12“:



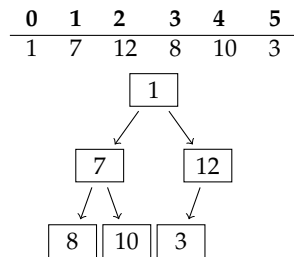
Nach dem Einfügen von „8“:



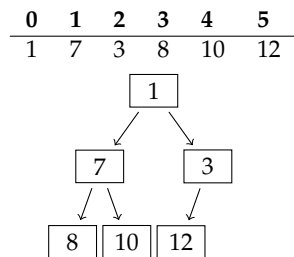
Nach dem Einfügen von „10“:



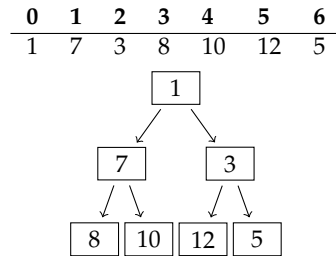
Nach dem Einfügen von „3“:



Nach Vertauschen von „3“ und „12“:



Nach dem Einfügen von „5“:



- (b) Was ist die worst-case Laufzeit in O-Notation für das Einfügen eines Elements in einen Heap der Größe n ? Begründen Sie ihre Antwort.

Die worst-case Laufzeit berechnet sich aus dem Aufwand für das Durchsickern eines eingefügten Elementes. Da das Durchsickern entlang eines Pfades im Baum erfolgt, entspricht der Aufwand im ungünstigsten Fall der Höhe des Baumes, d. h. $\log_2 n$. Insgesamt ergibt sich somit eine worst-case Laufzeit von $\mathcal{O}(\log n)$.