

**Aufgabe 6.1** (4 Punkte)

Gegeben sei ein leerer Heap.

- a) Fügen Sie die Elemente 16, 18, 9, 39, 12, 25, 8 und 5 in dieser Reihenfolge ein und stellen Sie diesen nach jedem Schritt grafisch dar. Stellen Sie den entstandenen Heap am Ende auch in einem Array dar.
- b) Entfernen Sie aus dem resultierenden Heap zweimal das Minimum und stellen Sie den sich ergebenden Heap jeweils dar.

**Aufgabe 6.2** (4 Punkte)

Zeigen Sie:

- a) Die im Algorithmus *siftDown* (Buch S. 166) angegebenen Zusicherungen beweisen die Korrektheit. Im Wesentlichen ist zu zeigen: Wenn nur  $h[j]$  eventuell zu groß ist, dann ist entweder  $h$  ein Heap oder die Operation  $swap(h[j], h[m])$  bewirkt, dass nur  $h[m]$  eventuell zu groß ist.
- b) Die eben beschriebene Implementierung von *siftDown* benötigt nicht mehr als  $2 \log n$  Schlüsselvergleiche.

**Aufgabe 6.3** (4 Punkte)

Beschreiben Sie einen Algorithmus, der  $k$  sortierte Listen mit Gesamtlänge  $n$  in  $O(n \log k)$  Zeit zusammenfügt zu einer sortierten Liste.

*Hinweis:* Benutzen Sie einen Heap der Größe  $k$ .