Theo III - Blatt 2

Benjamin Möller & Nick Daiber

November 12, 2024

1

a

```
func merge(l1, l2) {
    list res
    n = len(l1)
    m = len(l2)
    index1 = 0
    index 2 = 0
    while (index 1 < n and index2 < m) {
        // Größtes Element in res
        if (l1[index1] <= l2[index2]) {
            res[index1 + index2] = l1[index1]
            index1++
        }
        else {
            res[index1 + index2] = l2[index2]
            index2++
        }
    }
    // verbleibende Elemente anhängen
    if (index1 == n) {
        append(res, l2[index2+])
    }
    else {
        append(res, l1[index1+])
    }
    return res
}
```

Dabei gilt die Anzahl an Vergleichen $V \leq |l1|+|l2| = m+n \Rightarrow V \in O(n+m)$

b

```
func mergeLists(Number[][] lists, left, right) {
    if (left == right) {
        return lists[left]
    }
    mid = ⌊ left + (right - left) / 2 ⌋

    Number[] leftMerged = mergeLists(lists, left, mid)
    Number[] rightMerged = mergeLists(lists, mid + 1, right)

    return merge(leftMerged, rightMerged)
}
```

c

- Die delete-Routine entfernt ein gegebenes Element aus dem Baum, indem sie das Element in $O(n)$ Operationen findet, löscht und das letzte Element des Baumes an dessen Stelle schreibt und dann wieder für die Heapeigenschaft sorgt.
- Die change-key-Routine ändert den Wert für ein gegebenes Element aus dem Baum, indem das Element gefunden wird, dessen Wert überschrieben wird und zuletzt die Heapeigenschaft wieder hergestellt wird.

d

Für einen ternären Heap gilt v Elternknoten mit kindern s_0, s_1, s_2.
$s_0 = 3 \cdot v + 1, s_1 = 3 \cdot v + 2, s_3 = 3 \cdot v + 3$

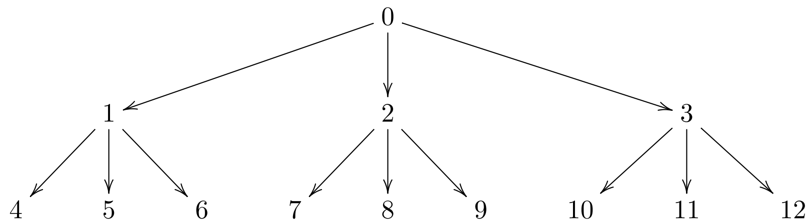

Figure 1: Darstellung ternärer Heap als Baum mit Index

2

a

Größtes Element wird nach oben durchgereicht.
Induktion über n für $n = 1$ trivial für $n = n + 1$ ist die liste für $j = n$ sortiert $n + 1$ wird ”durchgereicht”

b

best case

A bereits sortiert, nur $j = 1$ bis $n - 1 \Rightarrow n - 1$ vergleiche

worst case

A ”absteigend sortiert für $j = 1$ bis $n-1$ jeweils $j-1$ vergleiche, also $\frac{(n-1)(j-1)}{2}$ vergleiche

3