8. Tutorium - Algorithmen I

Nina Zimbel

10.06.2015

Heute:

- Bulk Insertion bei binären Heaps
- Suchbäume
- (k-Wege Merging)

Bulk Insertion bei binären Heaps

Gegeben sei ein binärer Heap, der n Elemente enthält. Es sollen nun k Elemente auf einmal eingefügt werden. Geben Sie ein Verfahren an (kein Pseudocode), mit dem man das Einfügen in

$O(\min\{k \log k + \log n,\ k + \log n \log k\})$

Schritten erledigen kann.
Sie können davon ausgehen, dass der Heap genau $2m - 1$ Elemente enthält ($m \in \mathbb{N}$).

- **Fall 1:** $k \in \Omega(n)$

- **Fall 1:** $k \in \Omega(n)$
- rufe *buildHeap* auf allen $n + k$ Elementen auf: $O(k)$
- **Fall 2:** $k \in o(n)$

- **Fall 1:** $k \in \Omega(n)$
- rufe *buildHeap* auf allen $n + k$ Elementen auf: $O(k)$
- **Fall 2:** $k \in o(n)$
- **Fall 2.1:** $k \leq \log n$

- **Fall 1:** $k \in \Omega(n)$
- rufe *buildHeap* auf allen $n + k$ Elementen auf: $O(k)$
- **Fall 2:** $k \in o(n)$
- **Fall 2.1:** $k \leq \log n$
- also brauchen wir die Laufzeit: $O(k \log k + \log n)$

- **Fall 1:** $k \in \Omega(n)$
- rufe *buildHeap* auf allen $n + k$ Elementen auf: $O(k)$
- **Fall 2:** $k \in o(n)$
- **Fall 2.1:** $k \leq \log n$
- also brauchen wir die Laufzeit: $O(k \log k + \log n)$
- **Fall 2.2:** $k > \log n$
- also brauchen wir die Laufzeit: $O(k + \log n \log k)$

Binäre Suchbäume

Operationen:

- locate
- insert

Suchbaum wird so beliebig unbalanciert.
Nächstes Tut: balancierte Suchbäume mit den Operationen split, fuse, balance

k-Wege Merging

Gegeben seien k doppelt-verkettete sortierte Listen $L_1, ..., L_k$ jeweils der Länge $\frac{n}{k}$. Geben Sie einen Algorithmus an, der in der Zeit

a) $\Theta(nk)$
b) $O(n \log k)$

eine sortierte Liste L erzeugt, die genau die Elemente der k sortierten Listen $L_1, ..., L_k$ enthält. Begründen Sie die Laufzeit.