Heapsort

Tobias Guggenmos

27. Januar 2016

Input: Ein unsortiertes Array
Output: Ein sortiertes Array

1 Heap

Definition 1 *Als max**Heap** bezeichnet man einen Binärbaum, bei dem jeder Elternknoten grösser als seine Kinder ist.*

In einem Heap gilt folgendes:

- Die Wurzel ist der grösste Knoten
- Alle Teilbäume eines Heaps sind ebenfalls Heaps

2 Algorithmus

Idee:

1. Man betrachtet das Inputarray als Baum und sortiert es zu einem Heap um
2. Der erste Eintrag des Arrays (also die Wurzel des Baums) ist nun das Grösste Element des Arrays. Diesen Vertauscht man mit dem letzten Eintrag des Arrays und schliesst ihn aus dem Heap aus.
3. Man repariert den Verbleibenden Baum wieder zu einem Heap und geht zu Punkt 2

Algorithmus zum Reparieren eines Heaps, der bis auf die Wurzel w passt:

1. Wenn w keine Kinder hat: fertig
2. Wenn beide Kinder von w kleiner als w sind: fertig
3. Sonst: Tausche w mit dem grössten Kind $k \rightarrow$ Die Wurzel ist wieder der grösste Knoten im Baum
4. Da k ersetzt wurde, kann es sein, dass der Teilbaum mit Wurzel k repariert werden muss $\rightarrow w = k$; fange wieder bei 1 an.

Algorithmus zum Umsortieren eines Arrays zu einem Heap

1. Repariere zuerst alle Teilbäume der letzten Elternreihe.
2. Die Teilbäume der vorletzten Elternreihe sind jetzt bis auf ihre Wurzel Heaps, repariere auch diese
3. Die Teilbäume der vorvorletzten ... bis zur Wurzel.

3 Effizienz

Grobe Komplexitätsabschätzung: Anzahl der Aufrufe von $siftDown$:

$$n_{Eltern} + n$$

- Unabhängig von der Sortierung
- Dauert also immer gleich lang
- **Kein Worst Case**
- In place
- Nachteil: Array muss wegen den vielen Tausche im RAM sein.

Quellen

`https://en.wikipedia.org/wiki/Heapsort`

Der Quellcode ist unter der MIT-License von mir auf github veröffentlicht
`https://github.com/slartibartfas/heapsort`