

Algorithmen und Datenstrukturen

B0. Pep-talk: Einführung in Datenstrukturen

Marcel Lüthi and Gabriele Röger

Universität Basel

31.03.2022

Datenstrukturen

Bad programmers worry about code. Good programmers worry about data structures and their relationships.

Linus Torvalds (Entwickler des Linux kernels)

Datenstrukturen

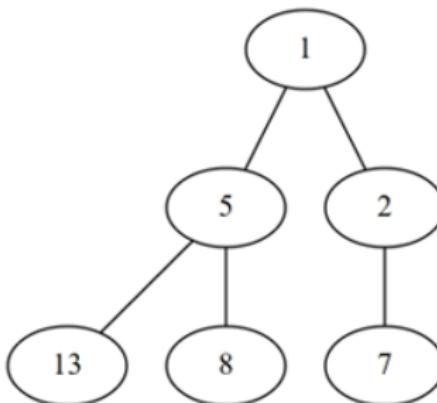
Show me your flowcharts and conceal your tables, and I shall continue to be mystified. Show me your tables, and I won't usually need your flowcharts; they'll be obvious

Fred Brooks (SW-Ingenieur & Gewinner Turing-Award)

Datenstruktur Heap

Heap

Ein (binärer) min-Heap ist ein vollständiger binärer Baum, bei dem gilt, dass der Wert in jedem Knoten kleiner gleich dem Wert seiner beiden Kindern (sofern vorhanden) ist.



Aufbauen eines Heaps

Elemente: 7, 13, 5, 8, 1, 2

Entfernen des kleinsten Elements vom Heap

Beispiel: Sortieren mit Heaps (Ausblick)

Idee des Algorithmus:

- Baue Heap aus unsortierter Liste
- Solange Elemente im Heap sind
 - Entferne kleinstes Element (Wurzel)
 - Schreibe Element in (neue) Liste
 - Stelle Heapbedingung wieder her
- Neue Liste enthält Elemente in sortierter Reihenfolge

Heapsort: Gleiche Idee, aber inplace.

Sortieren mit Heaps

Offene Fragen:

- Wie schnell können wir Heap aus n unsortierten Elementen aufbauen?
- Wie schnell können wir Heapbedingung nach Entfernen wiederherstellen?
- Wie gross ist die gesamte Laufzeitkomplexität

Sortieren mit Heaps

Offene Fragen:

- Wie schnell können wir Heap aus n unsortierten Elementen aufbauen?
- Antwort: Naiv: In $O(n \log_2 n)$ Operationen. Trickreich: In $O(n)$
- Wie schnell können wir Heapbedingung nach Entfernen wiederherstellen?
- Antwort: In $O(\log_2 n)$ Operationen
- Wie gross ist die gesamte Laufzeitkomplexität
- Antwort: In $O(n \log_2 n)$ Operationen

Komplexität verschoben von Algorithmus nach Datenstruktur

Datenstrukturen und Effizienz

Die richtige Datenstruktur macht den Code effizient:

jupyter Untitled (autosaved)

File Edit View Insert Cell Kernel Help Python [Root] O

Code CellToolbar

Algorithmen und Datenstrukturen

Interaktive Experimente

In [3]: `spyplot inline`
Populating the interactive namespace from numpy and matplotlib

In [7]: `plot(linspace(0, 1000), (linspace(0,1000) **2))`
Out [7]: [`<matplotlib.lines.Line2D at 0x29d8be022e0>`]

x	y
0	0
200	40000
400	160000
600	360000
800	640000
1000	1000000

Jupyter Notebook: datenstrukturen.ipynb

Zusammenfassung

- Algorithmen und Datenstrukturen arbeiten zusammen
 - (Teil der) Komplexität kann verschoben werden
- Datenstrukturen können meist visualisiert/graphisch verstanden werden
- Oft gilt: Gute Datenstrukturen \Rightarrow Einfach(ere) Programme

Details von Heapsort folgen ...