

## Wrap Up - Teil 1

Sebastian Millius, Sandro Feuz

Einige Verständniss Kontrollfragen und Aufgaben

**Analyse** Algorithmen können in einer sprach- und maschinenunabhängig analysiert und verstanden werden. Hierzu dient beispielsweise das Einheitskostenmodell und die asymptotische Analyse

- Was für Annahmen trifft das Einheitskostenmodell? Wann ist dieses Modell realistisch?
- Was sind Stärken, Schwächen?
- Was wäre ein alternatives Kostenmass? Wann ist dies angebracht?

Die asymptotische Worst-Case Analyse sind Hilfsmittel die den Vergleich der Effizienz von Algorithmen vereinfachen.

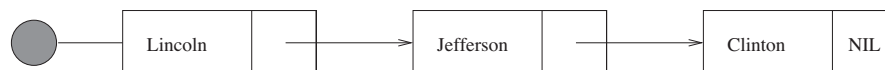
- Zeige dass die Big-O Relation transitiv ist, d.h. ist  $f(n) = \mathcal{O}(g(n))$ , und  $g(n) = \mathcal{O}(h(n))$ , dann ist  $f(n) = \mathcal{O}(h(n))$ .
- Die Stärke von Binärer Suche kommt von der logarithmischen Komplexität, nicht von der konkreten Basis des Logarithmus: Wie viele Abfragen braucht Binary Search um eine Telefonnummer im eine Million Einträge grossen Telefonbuch von Manhattan zu finden, wenn wir immer  $1/3$  und  $2/3$  anstatt  $1/2$  und  $1/2$  teilen? Vergleiche dies mit linearer Suche

### Elementare Datenstrukturen

- *Zusammenhängend allozierte Datenstrukturen* bestehend aus nebeneinanderliegenden Speicherzellen und beinhalten beispielsweise Arrays, Matritzen, Heaps und Hash Tables.

#### Eigenschaften

- *Konstante Zugriffszeit auf die Elemente*: da der Index eines Elements direkt auf eine Speicheradresse abgebildet werden kann.
- *Speichereffizienz*: keine zusätzlicher Speicher ist notwendig für beispielsweise Verlinkung
- *Speicherlokalität*
- Aufwändig, wenn die Grösse sich ändern muss: kann aber effizient mit Hilfe von *dynamischen Arrays* implementiert werden
- *Verlinkte Datenstrukturen* bestehen aus unabhängigen Einheiten, die durch Zeiger etc. verlinkt werden, beispielsweise Listen, Bäume etc.



Beispiel einer verlinkten Liste

- Was sind die (relativen) Vorteile/Nachteile einer Linked-List gegenüber einem Array?
- Was ist die Zugriffszeit auf ein beliebiges Element in einer Linked-List?

Datenstrukturen Design muss die unterstützten Operationen gegeneinander abwägen. Die schnellste Struktur die sowohl die Operationen  $A$  und  $B$  unterstützt, kann möglicherweise nicht die schnellste Struktur sein für nur  $A$  oder nur  $B$ .

Was ist die asymptotische Worst-Case Laufzeit für die Wörterbuchoperationen (Suche, Einfügen, Löschen, Nachfolger, Vorgänger, Minimum, Maximum) wenn die Datenstruktur implementiert ist als

- Unsortiertes Array
- Sortiertes Array
- Einfach verkettete unsortierte Liste
- Doppelt verkettete unsortierte Liste
- Einfach verkettete sortierte Liste
- Doppel verkettete sortierte Liste

Versuche folgende Fragen zu beantworten

- Wozu kann man einen Heap brauchen?
- Welche Operationen unterstützt ein Heap?
- Vergleiche und unterscheide Heap und Binärer Suchbaum
- Was ist die asymptotische Worst-Case Laufzeit für die Priority Queue Operationen (Einfügen, Find-Minimum, Delete-Minimum) wenn implementiert als
  - ein unsortiertes Array
  - sortiertes Array
  - Balancierter binärer Suchbaum
  - Heap

Eine kleine Zusammenfassung verschiedener Linked-List Probleme ist auf <http://cslibrary.stanford.edu/105/> verfügbar.

### Binäre Suchbäume

- Wozu braucht man Binäre Suchbäume
- Was ist der Vorteil von AVL gegenüber "normalen" Binären Suchbäumen
- Welche Strukturbedingung muss für einen AVL Baum gelten?
- Schreibe eine Funktion, die für einen gegebenen Binären Baum entscheidet, ob er ein AVL Baum ist. Was muss überprüft werden?
- Vergleiche Balancierte Binäre Suchbäume und Hash-Tabellen

**Sortieren** ist ein grundlegendes Verfahren und viele interessante Ideen und Ansätze des Algorithmen Designs tauchen auch im Zusammenhang mit Sortieren auf, beispielsweise Divide-and-Conquer, Datenstrukturen und randomisierte Verfahren. Historisch gesehen haben Computer die meiste Zeit sortiert: Ein Viertel aller Mainframe Cycles wurde zum Sortieren gebraucht (vgl. D. Knuth. The Art of Computer Programming, Volume 3: Sorting and Searching)

- Wenn Ihr eine Million Integer zu sortieren habt, welchen Algorithmus verwendet Ihr? Was ist die Laufzeit und der Speicherverbrauch?
- Beschreibe Vor- und Nachteile der bekanntesten Sortieralgorithmen.
- Entwerfe einen Algorithmus der ein Array nimmt und alle Duplikate entfernt, so dass nur noch distinkte Elemente vorkommen.
- Entwerfe einen Stack, der die Operationen Push, Pop (oberstes Element entfernen) und Minimum (der Elemente auf dem Stack) abfragen in konstanter Zeit ermöglicht.