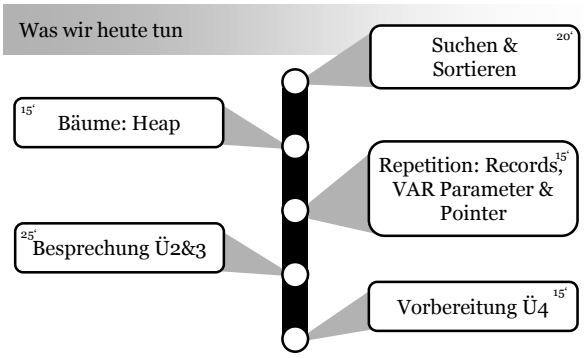
Informatik I WS 02/03

Übungsserie 3 per Email abgegeben?

Was wir heute tun



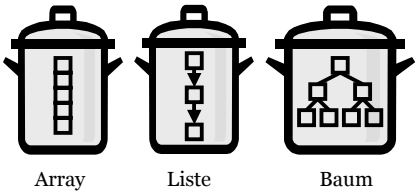
Auf der Suche nach Daten

Programme sammeln oft Daten,
welche sie in einen Topf schmeissen

Aber wie findet man die Dinge in diesem Topf?

Datenstrukturen zur Suche

Unser Topf hat eine innere Struktur,
welche die Suche ermöglicht



Array Liste Baum

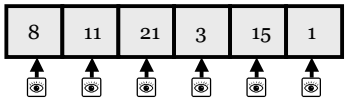
verschiedene Datenstrukturen für
verschiedene Aufgaben

Sortierte Daten

sortierte Daten besser zur Suche

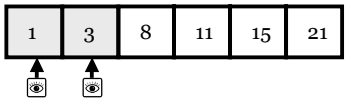
„ich will wissen, ob die 2 in der Liste steht“

unsortiertes Array:



„ich muss alles ansehen!“

sortiertes Array:



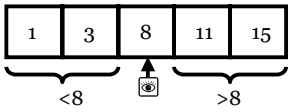
„nachdem ich die 3 seh‘, weiss ich, dass keine 2 mehr kommen wird!“

„Divide & conquer“ & binäre Suche

mit der Sortierung kann ich an einer Stelle auf den Rest der Daten schliessen

Suche wie vorhin relativ effizient
Binäre Suche/binary search ist aber noch besser!

Idee:



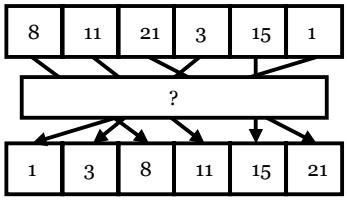
sortiertes Array

1. Schau in der Mitte des Arrays
2. ist es, was ich suche? wenn ja, dann fertig!
3. wenn nein, mach Schritt 1 mit dem in Frage kommenden Teilarray

} Divide & conquer

Wie sortieren?

sortierte Daten besser, aber wie sortiere ich diese?



unsortiert

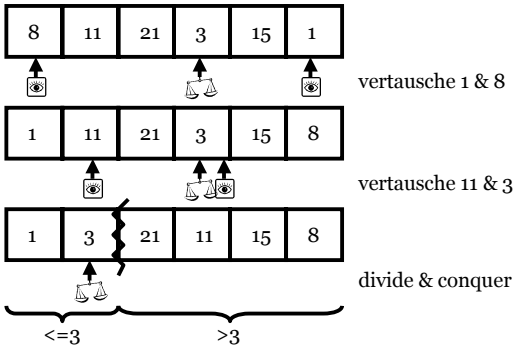
sortiert

sortierte Daten besser, aber wie sortiere ich diese?

beste Methode im Allgemeinen: Quicksort

Quicksort

sortierte Daten besser, aber wie sortiere ich diese?



Quicksort in Pseudocode

wähle das mittlere Element der Folge als x;

setze i = 0 und j = n-1;

wiederhole solange i <= j

suche von links das erste Element a_i mit a_i >= x;

suche von rechts das erste Element a_j mit a_j <= x;

falls i >= j

vertausche a_i und a_j;

setze i = i+1 und j = j-1;

aus http://www.iti.fh-flensburg.de/lang/algorithmen/sortieren/quick/quick.htm

Besprechung Serie 2 Übung 1

	gibt Zahl zurück	gibt Boolean zurück	kann beides zurückgeben
braucht Zahlen	`+ -` `* DIV`		`a[...]`
braucht Boolean		`~` `&` `OR`	
kann beides verwenden		`# =` `< >`	

Was ist Semantik?

Was ist Syntax?

Beprechung Serie 2

VAR Parameter: Immer Programmfluss aufschreiben!

Die Swap-Funktion

Seiteneffekte: Durchgeführte Operationen auflisten

TYPE Deklaration für Komplexe Zahl, Vektor & Matrize

Flugplan: Adjazenzmatrix versus Records

Datenstruktur modellieren... Annahmen, Einschränkungen, Implementation

Stammbaum: auf Eltern zeigen, nicht auf Kinder
Geschwister mit Ringliste

Beprechung Serie 3

Rekorde:
wenn bisheriges Maximum überschritten wird, eins dazu addieren

```
IMPORT RandomNumbers;
RandomNumbers.Uniform();
```

Wahrscheinlichkeitstheoretisch:
reverse-for all days i
value := value + 1.0 / i

Auskunft:

```
PROCEDURE frage(tag : INTEGER; summe : REAL): INTEGER;
```

Frage Antwort

jederzeit abrufbare Datenstruktur:
zusätzliches Indexarray

Vorbereitung Übung 4

Quicksort, Prinzip Divide & conquer, Median?

Dateien einlesen mit Modul `Files`

```
IMPORT Files;

PROCEDURE bla*();
VAR f : Files.File;         (* enthält später die datei *)
    r : Files.Rider;       (* enthält später den cursor *)
    c : CHAR;
BEGIN
      f := Files.Old(„EinDateiname");        (* öffnen *)
      Files.Set(r,f,0);                (* rider setzen *)
      Files.Read(r,c);          (* erstes zeichen lesen *)
      WHILE ~r.eof DO          (* solange nicht am ende *)
            Files.Read(r,c);           (* lies zeichen *)
            Out.Char(c);
      END;
END bla;
```

aus http://www.cs.inf.ethz.ch/37-001/faq.html