

Übungsserie 4

- Email, wenn Testatpunkt gewünscht
- Gedruckt, wenn Korrektur gewünscht

Was wir heute tun

15' Hash Tabellen

25' Besprechung Ü3+4

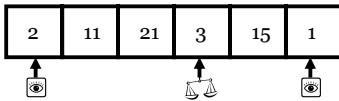
Quicksort ^{5'}

Abriss:
Datenbanken
& Funktionen darauf ^{10'}

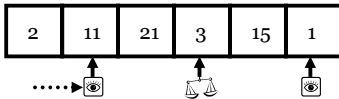
Vorbereitung Ü5 ^{15'}

Quicksort Schritt für Schritt 1-3

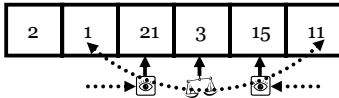
1 Waage setzen
Augen setzen



2 Nach innen
laufen bis relativ
falsch zur Waage

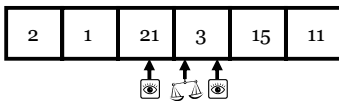


3 Elemente
Vertauschen &
nach innen wie 2

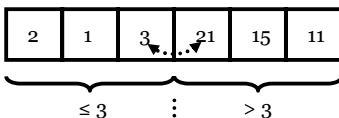


Quicksort Schritt für Schritt 4-6

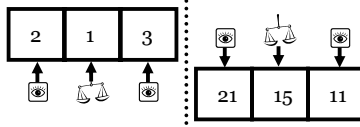
4 Wie in 2 nach
innen weiter



5 Vertauschen &
Waage wurde
Ausgetauscht...

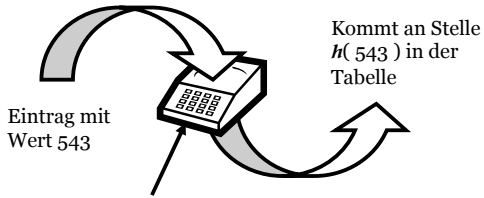


6 ...darum
Aufteilen und
Rekursion...



Hash Tabellen

Idee: Berechnung des Speicherortes aus Inhalt

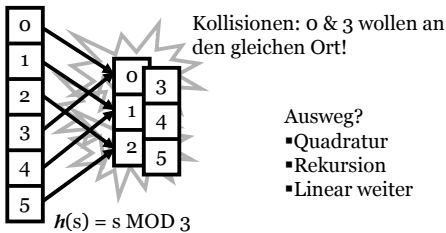


Vorteil: Gesuchtes/Suchabfrage enthält selbst den Speicherort, dadurch keine Suche nötig

Hash Funktion & Probleme

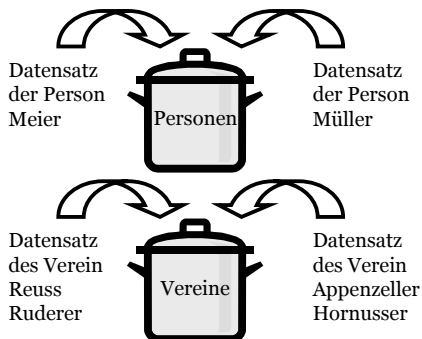
$$h(s) = s \text{ MOD } n$$

Problem: nicht injektiv, also gibt es mehrere s die an die Stelle $h(s)$ wollen:



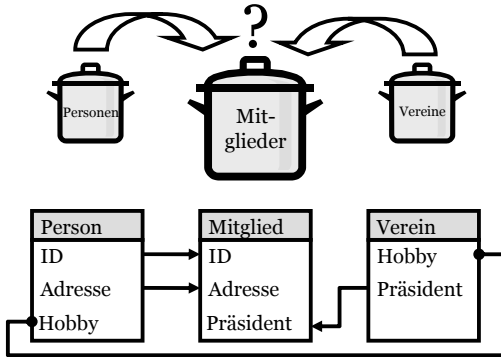
Abriss: Datenbanken

Datenbank ist auch ein Kochtopf



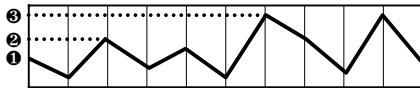
Eintopf oder Join-Funktion

Was passiert beim Eintopf?



Besprechung Übungsserie 3 Aufgabe a

Was ist ein Rekord?



Wie finden wir die Anzahl Rekorde?

```
Anzahl := 1;  
Maximum := erster Wert;  
Gehe von links durch ganzes Array  
  teste, ob (i-ter Wert) > Maximum;  
  wenn grösser, dann  
    Maximum := (i-ter Wert);  
    Anzahl := Anzahl + 1;  
Resultat := Anzahl;
```

Besprechung Übungsserie 3 Aufgabe b

Was ist die Wahrscheinlichkeit, dass ein Rekord kommt?

Vor Anfang:
 $p_1 = 1$, weil auf jeden Fall erstes Element ein Rekord ist

Danach:
Wahrscheinlichkeit wird kleiner, weil Rekord tendenziell höher, $p_{i>1} \sim 1/i$, bei uniformer Verteilung

Lösung:
Schätzung := 0;
Gehe durch ganzes Array
 Schätzung := Schätzung + (1 / (i-ter Wert));
Resultat := Schätzung;

Besprechung Übungsserie 3 Aufgabe c

Frage: Nach wievielen Tagen ab gegebenem Tag hat die Regenmenge eine bestimmte Schranke überschritten?

Auskunft:

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

Lösung:

```
i := tag;
```

während $schranke \leq 0$ wiederhole

```
    schranke := schranke - (i-ter Wert);
```

```
    i := i + 1;
```

```
Resultat := i - tag;
```

Variante:

Alle akkumulierten Werte in einem Array errechnen;

Binary Search im Teilarray links vom gegebenen Tag

nach ($akkumuliert[tag] + schranke$)

Besprechung Übungsserie 3 Aufgabe d

Wie kann ich diesen riesigen Array schlau sortieren?

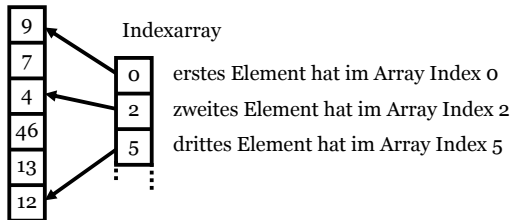
Heute wisst Ihr mit:

QuickSort

Damals hättet Ihr gewusst:

Index-Tabelle,
externes Sortieren

Datenarray



Aber eigentlich zu umständlich

Vorbereitung Übungsserie 5 Aufgabe 1

Konversion? = Umwandlung

„blabla“ wird z.B. zu „123133122123133122“

Macht das Sinn? Jawohl, der Computer macht das ohnehin

Jeder Buchstabe und Schriftzahl

ist eine Zahl zwischen 0 und 255

```
zahl := ORD( buchstabe );
```

Durch den ARRAY OF CHAR düsen: letztes Zeichen: 0x

Mögliche Fehler:

- Buchstaben sind drinnen

- Zu grosse Zahl: MAX (LONGINT)

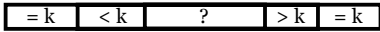
Vorbereitung Übungsserie 5 Aufgabe 2

Mischen - Zufallsstrategie

Das Element wird zufällig ausgewählt...

wähle zufälliges Element k aus Teilarray
wiederhole

- geh' mit l . Auge von links her nach innen, bis $\geq k$
- geh' mit r . Auge von rechts her nach innen, bis $\leq k$
- wenn l and r vorbei, dann raus aus wiederhole
- wenn $l > k$ und $r < k$, dann tausche l und r
- wenn $l > k$ und $r = k$, dann hänge r ans linke Ende
- wenn $l = k$ und $r < k$, dann hänge l ans rechte Ende
- wenn $l = r = k$,
dann hänge l ans linke und r ans rechte Ende



Vorbereitung Übungsserie 5 Aufgabe 3

Schlaue Hash-Funktion für Strings?

Habt Ihr in der Aufgabe 1 gemacht;
jetzt noch MOD MyHashTableArraySize anhängen

Aber Achtung – Kollisionen noch behandeln

Register:

1. Array mit Wörtern aus Kommandozeile
2. Array durchgehen
Hash-Wert des Wortes berechnen;
an der Stelle Wort in Hash-Tabelle schreiben;

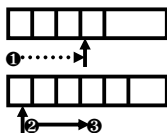
Search:

1. Wörter der Datei durchlaufen
jeweils Hash-Wert des Wortes berechnen;
mit Out ausgeben, falls $h(\text{Wort}) = \text{Wort}$;

Vorbereitung Übungsserie 5 Aufgabe 4

Dateien in zwei Arrays mit Records einlesen
Beide Arrays intelligent durchgehen
wenn $A.Destination = E.Source$
dann neues Record in drittem Array;
wenn $E.Destination = A.Source$
dann neues Record in drittem Array;
Resultat := drittes Array;

Was heisst intelligent? Was lässt sich ausschliessen?



sind Einzelschritte ② nötig, oder schon getestet, ① besser?
