Suchen & Sortieren

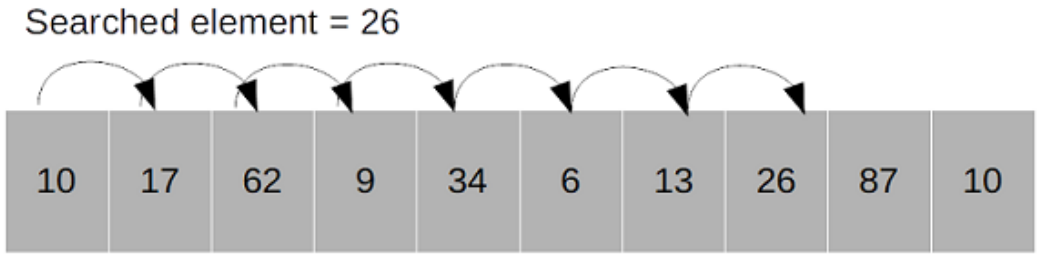
- Sequenzielle Suche
- Binäre Suche
- Duplikate im Array
- Sortierverfahren
- Selection Sort
- Bubble Sort

Suchen: Motivation

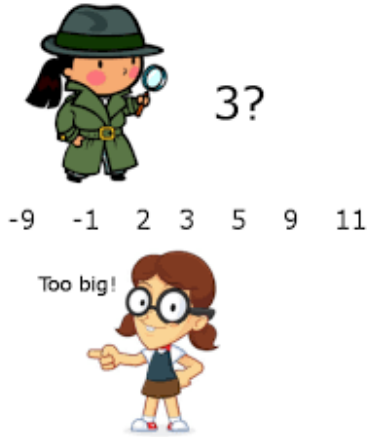
- Eine der häufigsten Aufgaben in der Informatik
- Einfache Fälle: Suchen in sortierten / unsortierten Folgen
- Vereinfachende Annahmen:
  - Folge = Feld von numerischen Werten
  - Auf jedes Element der Folge **F** kann über den Index **i** zugegriffen werden
    Erstes Element: **F[0]**, letztes Element bei n Werten: **F[n-1]**
  - Für die Feldelemente sind die bekannten Vergleichsoperatoren
    =, < und > definiert
  - Der für die Suche relevante Teil ist der numerische Wert, nach dem gesucht wird → **Suchschlüssel**

Sequenzielle Suche

- Die Folge wird sequenziell durchlaufen, beginnend beim ersten Element
- In jedem Schritt wird das aktuelle Element mit dem Suchschlüssel verglichen
- Sobald das gesuchte Element gefunden wurde, ist die Suche beendet
- Wenn das Ende der Folge erreicht wird, ohne das Suchelement gefunden zu haben, wird als Ergebnis ein spezielles Element oder ein ungültiger Index zurückgegeben



Binäre Suche



- Folge ist sortiert
- gesuchtes Element wird mit mittlerem Element der Folge verglichen
- bei Ungleichheit wird nur die jeweils linke oder rechte Hälfte der Folge weiter betrachtet
- jeder Vergleich teilt den Suchraum in zwei Hälften
  - bis das Element gefunden wurde
  - bis der verbleibende Suchraum sich nicht weiter unterteilen lässt

Vergleich: k==a[m] mit m = (0+n) / 2

Fall 1: k==a[m] → fertig

Fall 2: k < a[m] → rekursive Suche von k in a[0..m-1]

Fall 3: k > a[m] → rekursive Suche von k in a[m+1..n]

Sortieren: Motivation

- ca. 25% der kommerziell verbrauchten Rechenzeit
- triviale Aufgabe, viele Verfahren
- Vergleichskriterien:
  - Lösungsstrategie
    - Greedy
    - Divide & Conquer
    - Rekursion
  - Effizienz
    - Laufzeit, Speicherbedarf
  - intern / extern
  - stabil / instabil
  - allgemein / spezialisiert
    - vergleichsbasiert
    - bestimmte Schlüssel-Struktur

Gruppendiskussion

Sortieren von Spielkarten
(z.B. Skatspiel – französisches Blatt).

***Ziel:** Formulierung eines Algorithmus in Pseudocode zum Sortieren einer Folge von Spielkarten der Länge n.*

***Vorher:** Definition der Rahmen-bedingungen, d.h. wodurch ist die Sortierreihenfolge gegeben. Definieren Sie eine „größer“-Beziehung.*

Selection Sort

- Kann auch beim Sortieren von Spielkarten angewandt werden
- Sortieren einer Folge (Array, Stapel)
  - Suche das kleinste Element
  - Füge dieses Element am Anfang der Folge ein
  - Dies wird in jedem Schritt mit einem jeweils um 1 verkleinerten Bereich der Folge ausgeführt, solange bis die Folge die Länge 1 hat.
  - Somit sammeln sich am Anfang der Folge die bereits sortierten Elemente

Selection Sort: Beispiel

0	1	2	3	4	5	6	7	8	9	10	11	12	13
S	O	R	T	I	E	R	A	U	F	G	A	B	E

Bubble Sort

- Man vergleicht paarweise immer 2 benachbarte Elemente miteinander
- Entsprechen diese Elemente nicht der Sortierreihenfolge, dann vertauscht man sie
- Elemente, die größer sind als ihr Nachfolger, "überholen" diese und steigen zum Ende der Folge hin

Bubble Sort: Beispiel

0	1	2	3	4	5	6	7	8	9	10	11	12	13
S	O	R	T	I	E	R	A	U	F	G	A	B	E

Gruppendiskussion

Sortieren von Spielkarten
(z.B. Skatspiel – französisches Blatt),
wobei Sie nicht alle Spielkarten
gleichzeitig zugreifen (sehen) können.

***Ziel:** Formulierung eines Algorithmus in Pseudocode zum Sortieren einer Folge von Spielkarten der Länge n.*

- *Die Spielkarten liegen alle auf einem Stapel, der auf 2 Stapel verteilt werden kann.*
- *Für das Sortieren liegen die Spielkarten jetzt auf 2 Stapeln…*
- *Sie können immer nur die oberste Karte des Stapels sehen.*
- *Die Spielkarten können wieder auf einen (dritten) Stapel (zusammen)gelegt werden.*

- Insertionsort:
  - Nimm jeweils eine Karte vom Tisch und füge sie an der richtigen Stelle in die Hand ein.
- Selectionsort:
  - Suche jeweils die niedrigste Karte von denen, die auf dem Tisch liegen und füge sie rechts außen in die Hand ein
- BubbleSort:
  - Nimm die Karten auf die Hand und vertausche zwei benachbarte Karten, wenn sie in der falschen Reihenfolge sind. Tue das bis die Karten geordnet sind
- Mergesort:
  - Teile die Karten in zwei Teile. Sortiere die beiden Haufen einzeln und füge sie zusammen, wobei die Sortierung erhalten bleibt