

Einführung in die Programmierung II

Such- und Sortierv Verfahren

Reiner Hüchting

28. Februar 2022

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortierverfahren

Rekursive Sortierverfahren

Optimierungen beim Sortieren

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortierv Verfahren

Rekursive Sortierv Verfahren

Optimierungen beim Sortieren

Zusammenfassung zu Listen

Listen-Datentypen

- ▶ allgemeine Form des Arrays
- ▶ Zugriff auf Elemente über Index
- ▶ elementare Funktionen wie `push_back` etc.
- ▶ Implementierung unabhängig von konkreten Daten
 - ▶ deshalb meist nur `int` als Beispiel
- ▶ Konkreter Listentyp ist nur von technischer Bedeutung:
 - ▶ Unterschiede bei Performance, Speichernutzung etc.
 - ▶ im Idealfall keine Unterschiede bei Benutzung

Zusammenfassung zu Listen

Meist vielfältige Datensätze

- ▶ z.B. Adressbuch, Sensordaten, Figuren in Spielen ...

Zusammenfassung zu Listen

Meist vielfältige Datensätze

- ▶ z.B. Adressbuch, Sensordaten, Figuren in Spielen ...

Eigenschaften von Datenelementen

- ▶ Daten müssen vergleichbar sein
- ▶ Daten haben eine Ordnung!
 - ▶ einzelne Schlüssel oder gesamter Datensatz
 - ▶ Ordnung/Schlüssel vereinfachen Suchen und Sortieren.

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortierv Verfahren

Rekursive Sortierv Verfahren

Optimierungen beim Sortieren

Suchverfahren

Ziel: Herausfinden, ob und an welcher Stelle in einer Liste ein bestimmtes Element steht.

Suchverfahren

Ziel: Herausfinden, ob und an welcher Stelle in einer Liste ein bestimmtes Element steht.

Suchen in unsortierten Listen:

- ▶ Einzige Möglichkeit: Die Liste linear durchsuchen.

Suchverfahren

Ziel: Herausfinden, ob und an welcher Stelle in einer Liste ein bestimmtes Element steht.

Suchen in unsortierten Listen:

- Einzige Möglichkeit: Die Liste linear durchsuchen.

Suche in sortierten Listen: **Binäre Suche**

1. Vergleich mit mittlerem Element.
2. Ist das gesuchte Element kleiner, suche in linkem Teil weiter.
3. Ist das gesuchte Element größer, suche in rechtem Teil weiter.

Suchverfahren

Ziel: Herausfinden, ob und an welcher Stelle in einer Liste ein bestimmtes Element steht.

Suchen in unsortierten Listen:

- Einzige Möglichkeit: Die Liste linear durchsuchen.

Suche in sortierten Listen: **Binäre Suche**

1. Vergleich mit mittlerem Element.
2. Ist das gesuchte Element kleiner, suche in linkem Teil weiter.
3. Ist das gesuchte Element größer, suche in rechtem Teil weiter.

Die binäre Suche kann effizient **rekursiv** implementiert werden.

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortiervverfahren

Rekursive Sortiervverfahren

Optimierungen beim Sortieren

Einfache Sortiervverfahren

Insertionsort

- ▶ „Sortieren durch Einfügen“
 - ▶ Durchlaufe die Liste einmal.
 - ▶ Sortiere dabei jedes Element in eine neue Liste ein.
- ▶ Die Idee kann In-Place umgesetzt werden (d.h. in der vorhandenen Liste).

Einfache Sortierverfahren

Insertionsort

- ▶ „Sortieren durch Einfügen“
 - ▶ Durchlaufe die Liste einmal.
 - ▶ Sortiere dabei jedes Element in eine neue Liste ein.
- ▶ Die Idee kann In-Place umgesetzt werden (d.h. in der vorhandenen Liste).

Algorithmus:

1. Sortiere die ersten beiden Elemente.
2. Ordne jedes weitere Element in den unteren Bereich ein.

Einfache Sortiervverfahren

Selectionsort

- ▶ „Sortieren durch Auswählen“
 - ▶ Durchlaufe die Liste mehrfach.
 - ▶ Finde das kleinste Element und hänge es an eine neue Liste an.
- ▶ Auch dieses Verfahren kann In-Place umgesetzt werden.

Einfache Sortierverfahren

Selectionsort

- ▶ „Sortieren durch Auswählen“
 - ▶ Durchlaufe die Liste mehrfach.
 - ▶ Finde das kleinste Element und hänge es an eine neue Liste an.
- ▶ Auch dieses Verfahren kann In-Place umgesetzt werden.

Algorithmus:

- ▶ Durchlaufe die Liste n mal.
 1. Finde dabei jeweils das kleinste Element.
 2. Vertausche das Element mit dem ersten Element.
- ▶ Im i -ten Durchlauf betrachte nur die Elemente i bis n .

Einfache Sortierverfahren

Bubblesort

- ▶ Idee: Elemente steigen wie Blasen auf.
 - ▶ Vergleiche jeweils zwei benachbarte Elemente.
 - ▶ Wenn sie falsch sortiert sind, vertausche sie.

Einfache Sortierverfahren

Bubblesort

- ▶ Idee: Elemente steigen wie Blasen auf.
 - ▶ Vergleiche jeweils zwei benachbarte Elemente.
 - ▶ Wenn sie falsch sortiert sind, vertausche sie.

Algorithmus:

- ▶ Durchlaufe die Liste:
 1. Vergleiche Element i mit $i + 1$.
 2. Vertausche die Elemente, falls sie falsch sortiert sind.
- ▶ Wiederhole den Vorgang, bis die Liste sortiert ist.
- ▶ Im j -ten Durchlauf betrachte nur die Elemente von 0 bis $n - j + 1$.

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortierv Verfahren

Rekursive Sortierv Verfahren

Optimierungen beim Sortieren

Rekursive Sortierverfahren

Quicksort

- ▶ Idee: Sortiere die Liste in **große Zahlen** und **kleine Zahlen** vor.
- ▶ Sortiere die Teillisten und hänge sie wieder zusammen.

Rekursive Sortierverfahren

Quicksort

- ▶ Idee: Sortiere die Liste in **große Zahlen** und **kleine Zahlen** vor.
- ▶ Sortiere die Teillisten und hänge sie wieder zusammen.

Algorithmus:

1. Wähle ein **Pivotelement**.
2. Erzeuge zwei (möglichst gleich große) Teillisten:
 - ▶ Elemente die **kleiner** sind als das Pivotelement.
 - ▶ Elemente die **größer** sind als das Pivotelement.
3. Sortiere rekursiv die beiden Teillisten.

Rekursive Sortierverfahren

Quicksort

- ▶ Idee: Sortiere die Liste in **große Zahlen** und **kleine Zahlen** vor.
- ▶ Sortiere die Teillisten und hänge sie wieder zusammen.

Algorithmus:

1. Wähle ein **Pivotelement**.
2. Erzeuge zwei (möglichst gleich große) Teillisten:
 - ▶ Elemente die **kleiner** sind als das Pivotelement.
 - ▶ Elemente die **größer** sind als das Pivotelement.
3. Sortiere rekursiv die beiden Teillisten.

Wie wählt man das Pivotelement?

- ▶ Mitte des Bereichs, mittleres Element oder erstes Element?
- ▶ Hängt von Struktur der Daten ab.

Rekursive Sortierverfahren

Mergesort

- ▶ Lösung ohne das Pivot-Problem:
Ohne Vorsortierung direkt die Liste zerlegen.
- ▶ Erst beim Zusammensetzen die Liste sortieren.

Rekursive Sortierverfahren

Mergesort

- ▶ Lösung ohne das Pivot-Problem:
Ohne Vorsortierung direkt die Liste zerlegen.
- ▶ Erst beim Zusammensetzen die Liste sortieren.

Algorithmus

1. Zerlege die Liste in gleich große Teillisten.
2. Sortiere die Teillisten rekursiv.
3. **Mische** die Teillisten zusammen.

Themenüberblick – Suchen und Sortieren

Zusammenfassung zu Listen

Suchverfahren

Einfache Sortierv Verfahren

Rekursive Sortierv Verfahren

Optimierungen beim Sortieren

Optimierungen beim Sortieren

Shellsort

- ▶ Problem bei Insertionsort: Elemente müssen zum Einsortieren oft über weite Strecken verschoben werden.
- ▶ Andererseits: Insertionsort arbeitet sehr effizient, wenn die Liste schon fast sortiert ist.

Optimierungen beim Sortieren

Shellsort

- ▶ Problem bei Insertionsort: Elemente müssen zum Einsortieren oft über weite Strecken verschoben werden.
- ▶ Andererseits: Insertionsort arbeitet sehr effizient, wenn die Liste schon fast sortiert ist.

Idee: Vorab eine grobe Ordnung herstellen.

- ▶ Betrachte zunächst nur jedes i -te Element.
- ▶ Verringere i in jedem Schritt bis $i = 1$.

Optimierungen beim Sortieren

Bucketsort

- ▶ Vor- oder Teilsortierung mittels Kategorien
- ▶ Oft ist der Datenbereich strukturiert, Elemente können kategorisiert werden.
- ▶ Vollständige Sortierung evtl. nicht notwendig.

Optimierungen beim Sortieren

Bucketsort

- ▶ Vor- oder Teilsortierung mittels Kategorien
- ▶ Oft ist der Datenbereich strukturiert, Elemente können kategorisiert werden.
- ▶ Vollständige Sortierung evtl. nicht notwendig.

Algorithmus:

1. Sortiere Daten nach Kategorien in sog. **Buckets**
 - ▶ Kategorien können Jahreszahlen, Anfangsbuchstaben, Zahlenbereiche etc. sein.
2. Sortiere jeden Bucket einzeln mit einem klassischen Verfahren.