

## Suchen

Nahezu jedes Informatiksystem speichert eine Menge an Daten, mit denen das System arbeitet. Diese Daten werden i.d.R. in Datenstrukturen (z.B. Arrays) verwaltet. Dort sollen sie aber nicht nur einfach abgelegt, sondern bei Bedarf auch möglichst schnell wieder gefunden werden. Den Datensatz, nach dem gesucht werden soll, nennt man Suchschlüssel.

### Lineare Suche in einem Array

Mithilfe der linearen (sequenziellen) Suche kann ein unsortiertes Array durchsucht werden. Die Suche beginnt beim ersten Element. Stimmt dieses mit dem Suchschlüssel überein, kann die Suche beendet werden. Andernfalls fährt man mit dem zweiten Element fort. Die Suche läuft solange, bis das gesuchte Element gefunden wurde oder bis das Ende des Arrays erreicht wurde. Dann kann man sicher sein, dass das gesuchte Element nicht vorhanden ist.

```
public int [] feld = new int [50];

public boolean search(int n){
    for (int i = 0; i < feld.length; i++){
        if (feld[i] == n){
            return true;
        }
    }
    return false;
}
```

### Effizienzbetrachtung

Zur Bewertung der Effizienz eines Algorithmus betrachtet man seine Laufzeit, d.h. die Anzahl der benötigten Rechenschritte und seinen Speicherplatz. Die Laufzeit der linearen Suche hängt im Wesentlichen von der Menge der zu durchsuchenden Daten ab. Diese wird Problemgröße genannt und meistens mit  $n$  bezeichnet. Auch der Speicherbedarf für den Algorithmus der linearen Suche hängt von der Problemgröße ab. Die Eingabedaten verbleiben während der Ausführung des Algorithmus in ihrer Datenstruktur (dem Array), sodass kein zusätzlicher Speicher benötigt wird.

Um das Laufzeitverhalten eines Algorithmus zu beschreiben, werden nur die maßgeblichen Operationen betrachtet. Bei einem Suchalgorithmus sind das die Vergleiche.

Man unterscheidet 3 Fälle:

- best case
- average case
- worst case

### **Laufzeit lineare Suche**

|              |              |
|--------------|--------------|
| best case    | $T(n) = 1$   |
| average case | $T(n) = n/2$ |
| worst case   | $T(n) = n$   |