


## Suchen und Sortieren

### Suchalgorithmen

---

#### **Aufgabe 1**      **Lineare Suche I**

Kopiere das Projekt  01-Suchmaschine aus dem Tauschordner und öffne es in BlueJ.

Implementiere die Methode `public boolean lineareSuche(int suchzahl, int[] sucharray)`.

Die Methode soll im Array `sucharray` nach dem Verfahren der *linearen Suche* nach der Zahl `suchzahl` suchen. Ist die Zahl vorhanden wird `true` zurückgegeben, sonst `false`.

Du kannst deine Implementierung mit der Test-Klasse auf Fehler prüfen. Denk auch daran, dass du den *Debugger* zur Fehlersuche nutzen kannst.

#### **Aufgabe 2**      **Lineare Suche II**

Implementiere die Methode `public boolean lineareSuche(int suchzahl, List<Integer> suchliste)`.


Die Methode soll in der Liste `suchliste` nach dem Verfahren der *linearen Suche* nach der Zahl `suchzahl` suchen. Ist die Zahl vorhanden wird `true` zurückgegeben, sonst `false`.


#### **Aufgabe 3**      **Binäre Suche**

Implementiere die Methode `public boolean binaereSuche(int suchzahl, int[] sucharray)`.

Die Methode soll im Array `sucharray` nach dem Verfahren der *binären Suche* nach der Zahl `suchzahl` suchen. Ist die Zahl vorhanden wird `true` zurückgegeben, sonst `false`.

Du kannst davon ausgehen, dass das Array `sucharray` schon aufsteigend sortiert ist.

 **Tipp** Definiere Variablen für die untere (*von*) und obere (*bis*) Schranke des Suchbereichs. Zunächst sind sie auf 0 und `sucharray.length-1` gesetzt. Das mittlere Element liegt nun bei `(int)((von+bis)/2)`.

 **Tipp** Notiere dir in einer Tabelle, wie sich die Grenzen in einem kleinen Beispiel verändern und überleg dir, unter welcher Bedingung der Algorithmus jeweils beendet wird.

#### **Aufgabe 4**

- Begründe, warum die binäre Suche auf linearen Listen nicht schneller als die lineare Suche sein kann.
- Gibt es Fälle, in denen die lineare Suche schneller als die binäre ist? Finde Beispiele für Datensätze (Integer), bei denen dies der Fall ist und begründe deine Entscheidung.

#### ★ **Aufgabe 5**

Implementiere eine *rekursive* Variante der *binären Suche* (bzw. eine *iterative*, wenn du bei **Aufgabe 3** schon rekursiv implementiert hast).