

Suchen und Sortieren

Effizienz von Algorithmen

Die *Effizienz* eines Algorithmus ist nicht so einfach zu quantifizieren. Es gibt verschiedene Faktoren, die einen Algorithmus "gut" oder "schlecht" machen. Zwei wichtige Merkmale sind die *Laufzeit* und der *Speicherbedarf* des Algorithmus.

Die **Laufzeit** misst, wie "schnell" der Algorithmus ist. Da diese Messgröße von der Hardware beeinflusst wird, zählt man in der Regel, wie viele "elementare Operationen" der Algorithmus braucht.

Der **Speicherbedarf** misst, wie viel (zusätzlichen) Speicher der Algorithmus benötigt.

Um Algorithmen besser vergleichen zu können, betrachtet man drei unterschiedliche Fälle:

worst case: Wie Effizient ist der Algorithmus im schlechtesten Fall?

best case: Wie Effizient ist er im bestmöglichen Fall?


average case: Wie Effizient ist er im Mittel (über alle möglichen Fälle hinweg)?

Aufgabe 1

Finde Beispiele für die *best* und *worst cases* der linearen und binären Suche. Erkläre jeweils (umgangssprachlich), warum sie am schlechtesten sind.

Aufgabe 2

Zum Vergleich der Sortieralgorithmen wollen wir die *Anzahl der nötigen Vergleiche* in einem Zahlenarray der Größe n in den drei Fällen bestimmen.

 **Hinweis** Der best case ist jeweils trivial und kann ohne Messung bestimmt werden.

Implementiere Methoden nach dem Format `public int countWorstLinear(int n)`, die ein Array mit n Zufallszahlen erstellt und den schlechtesten Fall für die lineare Suche durchführt. Dabei soll die Anzahl an Vergleichoperationen gezählt und als Ergebnis zurückgegeben werden.

Erstelle jeweils Methoden für die Fälle der linearen und binären Suche.

 **Tipp** Das zufällige Zahlenarray kann durch `Zahlengenerator.zufallsArray(n)` erzeugt werden.

Aufgabe 3

Implementiere eine Testmethode, die für verschiedene n -Werte die Anzahl an Vergleichen zählt und die Ergebnisse auf der Kommandozeile ausgibt. Nutze dabei diese Ausgabe:

```
System.out.println(n+"\t"+count);
```

wobei n die Anzahl der durchsuchten Elemente ist und `count` die Anzahl der Vergleiche. `\t` erzeugt einen Tabulator zwischen den Zahlen.

Leg parallel ein neues Tabellenblatt in **LibreOffice Calc** an und notiere in der ersten Zeile die Überschriften n und *Vergleiche*. Hier kopierst Du die Ergebnisse der Messungen während des Versuchs hinein. Durch den Tabulator in der Ausgabe übernimmt **LibreOffice** die Werte direkt in die beiden Zellen.

Führe Messungen für verschiedene Eingabegrößen durch, zum Beispiel 100, 200, 300, ..., 1000, 2000, ..., 10000, ... und kopiere die Laufzeiten in das Tabellenblatt.

Aufgabe 4

Erstell ein Streudiagramm aus den Messergebnissen. Dabei soll n auf der x -Achse und die Vergleiche auf der y -Achse abgebildet werden. Wie kann das entstandene Diagramm interpretiert werden?