

3. Programmieraufgabe zur Informatik I

Abgabe vom 16.06. 10:00 Uhr bis zum 17.06. 17:00 Uhr
im Internet auf den Übungsseiten zur Vorlesung

In dieser Programmieraufgabe wollen wir uns mit den in der Vorlesung vorgestellten Suchalgorithmen auf sortierten sequentiellen Listen beschäftigen. Hierzu werden Sie in die folgenden Gruppen eingeteilt (Die Bearbeitung erfolgt weiterhin in **Einzelabgabe**):

A-K: Gruppe **Fibonacci-Suche**

L-Z: Gruppe **Exponentielle Suche**

Aufgabe 5:

Implementieren Sie eine Klasse `Suche`. Diese enthält eine Hauptroutine `public static void main(String[] args)`, sowie eine Methode `boolean search(int[] field, int key)`, die im Integer-Array `field` nach einem Element `key` sucht. Hierin soll die Fibonacci-Suche bzw. die exponentielle Suche implementiert werden.

Falls das Element `key` gefunden wird, gibt die Methode den booleschen Wert `true`, andernfalls den Wert `false` zurück. Die Hauptmethode gibt das entsprechende Ergebnis auf die Kommandozeile aus.

Beispiel-Werte

```
int field[5]={1, 2, 3, 4, 5}, int key=0
```

Beispiel-Ausgabe

```
Der Schluessel 0 ist nicht im Feld enthalten.
```

Zusätzlich können Sie selbstverständlich weitere (private) Methoden implementieren, sofern es notwendig bzw. nützlich ist.

Bei der Fibonacci-Suche kann dies z.B. eine Methode zur Bestimmung der n -ten Fibonacci-Zahl sein. (Vergleichen Sie eventuell die Laufzeit der rekursiven Bestimmung von F_n und der Berechnung von F_n mittels der geschlossenen Formel.)

Bei der exponentiellen Suche kann eine eigene Methode für die benötigte binäre Suche (rekursiv oder iterativ) implementiert werden.

Ihr Programm soll wie gehabt von der Kommandozeile einen Dateinamen übergeben bekommen. In dieser Datei steht ein sortiertes Feld, sowie eine Reihe zu suchender Schlüssel. In der ersten Zeile der Datei steht dabei die Anzahl der Feldeinträge n . Jede weitere Zeile (2te bis $n + 1$ te) enthält dann genau einen dieser ganzzahligen Einträge. Dieses Feld soll aus der Datei in ein `int []`-Arrays eingelesen werden.

Die $n + 2$ te Zeile enthält die Anzahl m der folgenden zu suchenden Schlüssel, die (wie die Feldeinträge) jeweils einzeln in den Zeilen $n + 3$ bis $n + m + 3$ stehen. Die zu suchenden Schlüssel können ebenfalls in ein `int []`-Arrays eingelesen und sollen dann sukzessive im ersten Feld gesucht werden.

Beispiel-Datei

```
5
1
2
4
5
6
2
3
6
```

Beispiel-Werte

```
int field[5] = {1, 2, 4, 5, 6}
int keyfield[2] = {3,6}
```