

4. Übungsblatt

Ausgabe: 03.05.2010 **Abgabe:** 10.05.2010, vor der Vorlesung

Aufgabe 1: Binäre Suche

10 Punkte

Es sei ein `int`-Array `f` folgenden Inhalts gegeben:

```
f[0] = 1;   f[1] = 2;   f[2] = 3;   f[3] = 5;  
f[4] = 7;   f[5] = 11;  f[6] = 13;  f[7] = 17;  
f[8] = 19;  f[9] = 23;  f[10] = 29; f[11] = 31;  
f[12] = 37; f[13] = 41; f[14] = 43; f[15] = 47;
```

Geben Sie an, wie die Methode `binarySearch` auf `f` arbeitet, wenn nach folgenden Elementen gesucht wird:

- (a) 13
- (b) 46

Geben Sie dabei für jeden Rekursionsaufruf die Werte der Parameter `l`, `r` und `p` an.

Aufgabe 2: Binäre Suche

10 Punkte

In einem Array kann ein Element mehr als nur einmal vorkommen. Wie würden Sie die Methode `binarySearch` abändern, damit stets die minimale Position eines gesuchten Elementes in einem sortierten Array ausgegeben wird, wenn es enthalten ist?

Aufgabe 3: Kombinationsverfahren

10 Punkte

In der Vorlesung haben wir für die beiden Methoden `linearSearch` und `binarySearch` die Komplexitäten im schlechtesten Fall zu

- $(k_1 + k_3) + k_2 n$ (für `linearSearch`)
- $(k'_1 + k'_2 + k'_3) \log_2 n + (k'_1 + k'_3)$ (für `binarySearch`)

bestimmt. Hierbei ist n stets die Größe des gegebenen Arrays.

Entwerfen Sie ein kombiniertes Suchverfahren, wenn die Konstanten $k_1 = k_2 = k_3 = 1$ sowie $k'_1 = k'_3 = 16$ und $k'_2 = 32$ sind. Bestimmen Sie dazu den Cross-over Point und implementieren Sie eine Methode `combinedSearch` in der Klasse `SearchClass`.