

VL11, Aufgabe 1 (Übung)

- Geben Sie zu den elementaren Suchverfahren an, welchen maximalen Aufwand sie haben und in welchen Fällen die jeweiligen Verfahren benutzt werden.
- Nachfolgend sind mehrere Zahlenfolgen gegeben. Suchen Sie mit dem jeweils geeigneten elementaren Suchverfahren nach der Zahl 32. Geben Sie an, welches Suchverfahren Sie verwenden. Beschreiben Sie zudem Ihr Vorgehen und geben Sie die Rechenschritte an. Verwenden Sie nach Möglichkeit jedes Verfahren mindestens einmal.

1.	2	5	11	15	20	32	50	160	300	456
----	---	---	----	----	----	----	----	-----	-----	-----

2.	30	72	36	12	44	59	27	32	8	66
----	----	----	----	----	----	----	----	----	---	----

3.	26	27	28	30	31	32	34	36	37	38
----	----	----	----	----	----	----	----	----	----	----

VL11, Aufgabe 2 (Übung)

Es sei ein ungeordnetes Array von n numerischen Schlüsseln (n sehr groß) gegeben. Es wird in diesem Array nach $n/2$ Schlüsseln gesucht. Vergleichen Sie die folgenden Möglichkeiten auf Grund ihrer maximalen asymptotischen Laufzeit:

- Sortieren mittels BubbleSort und das Suchen jeweils mit binärer Suche.
- Sortieren mittels HeapSort und das Suchen jeweils mit InterpolationsSuche.
- Suchen jeweils mit linearer Suche.
- Sortieren mittels MergeSort und das Suchen jeweils mit binärer Suche.

VL11, Aufgabe 3 (Übung)

Erstellen Sie eine Hash-Tabelle mit 11 Elementen. Benutzen Sie folgende Hash-Funktion für Zeichenketten:

- Jedem Buchstaben sei seine Position im Alphabet zugeordnet:
 $\text{ord}: \{A, B, \dots, Z\} \rightarrow \{1, 2, \dots, 26\}$
- $n = 11$
- $H(k) = (\text{ord}(1. \text{ Buchstabe von } k) + \text{ord}(2. \text{ Buchstabe von } k)) \bmod n$

Tragen Sie nacheinander die folgenden Schlüssel in der angegebenen Reihenfolge in die Hash-Tabelle ein: STARK, RETTINGER, KUKUK, KNABE, HOFFMANN und LAUENROTH.

- Die Auflösung von Kollisionen soll durch direkte Verkettung der Überläufer erfolgen.
- Die Auflösung von Kollisionen soll durch Quadratisches Sondieren erfolgen.

VL11, Aufgabe 4 (Übung)

Die Schlüssel 15, 38, 53, 75, 86, 100, 115 sind in dieser Reihenfolge in eine Hash-tabelle $t[0, \dots, 6]$ eingetragen worden:

115	15	86	38	53	75	100
-----	----	----	----	----	----	-----

Als Hashfunktion wurde $h(s) = s \bmod 7$ verwendet. Zur Kollisionsauflösung wurde das Quadratische Sondieren eingesetzt.

Suchen Sie in der Hashtabelle t nach den Schlüsseln 86 und 100. Geben Sie für jede Suche an, wieviele und welche Einträge in der Hashtabelle betrachtet werden mussten, um den Schlüssel zu finden.

VL11, Aufgabe 5 (Praktikum)

Erstellen Sie eine neues Projekt Ueb11-Suche mit einer Klasse Suche.

- a) Programmieren Sie in Java eine **iterative** Klassenmethode

```
static boolean binaereSuche(final String[] worte, final String begriff);
```

die in einem geordneten (sortierten) Array von Zeichenketten nach einer Zeichenkette sucht. Wird der gesuchte Begriff im Array gefunden, ist das Ergebnis `true`, ansonsten `false`. Nutzen Sie dabei die Methode `compareTo` der Klasse `String`.

- b) Testen Sie Ihre Methode mit einem geeigneten Hauptprogramm.

VL11, Aufgabe 6 (Praktikum)

Programmieren Sie die Methode `findePosition` der Klasse `HashTable` aus `UEB11-Hashtable.zip`. Zur Kollisionsauflösung soll quadratisches Sondieren verwendet werden.