

Übungen zur Vorlesung  
**Algorithmen und Datenstrukturen**  
WiSe 2019/20  
Blatt 9

Wichtige Hinweise:

- > Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!
- > Kursraum: <https://elearning.uni-regensburg.de/course/view.php?id=9228>

**Aufgabe 1:**

Ermitteln Sie die Positionen, an die die Schlüssel 61, 62, 63, 64, 65 platziert werden, wenn eine Hashtabelle der Größe  $m = 1000$  gegeben ist und die Werte mittels  $h(s) = \lfloor m \cdot ((s \cdot x) \bmod_c 1) \rfloor$  mit  $x = \frac{\sqrt{5}-1}{2}$  abgebildet werden.

**Aufgabe 2:**

Sei eine Hashtabelle der Größe  $m$  gegeben, in der  $n$  Schlüssel mittels offener Adressierung gespeichert werden sollen. Geben Sie der folgenden Formel eine sinnvolle Bedeutung:

$$P_k = \left(\frac{1}{m}\right)^k \left(1 - \frac{1}{m}\right)^{n-k} \binom{n}{k}$$

**Aufgabe 3:**

Implementieren Sie Hashing mit offener Adressierung in C, C++, C# oder Java anhand der Hashfunktion  $\hat{h}(s) = s$  unter Verwendung folgender Varianten zur Kollisionsauflösung:

1. Lineares Probieren
2. Quadratisches Probieren mit  $c_1 = 1$  und  $c_2 = 3$
3. Doppeltes Hashing mit  $h_1(s) = s$  und  $h_2(s) = 1 + (s \bmod (m - 1))$

Geben Sie jeweils die Hashtabelle der Größe  $m = 11$  aus, die nach Einfügen der Werte 10, 22, 31, 4, 15, 28, 17, 88, 59 in allen drei Varianten entsteht.

**Aufgabe 4:**

Realisieren Sie die Datenstruktur Skip-Liste in C, C++, C# oder Java. Jeder Schlüssel soll maximal einmal in einer Skip-Liste enthalten sein und folgende Operationen sollen mindestens zur Verfügung stehen:

- `Init()`: Initialisiert eine leere Skip-Liste
- `Deinit()`: Deinitialisiert eine Skip-Liste
- `Print()`: Gibt die aktuelle Skip-Liste aus (z.B. auf der Konsole)
- `Insert(key k)`: Fügt einen Schlüssel  $k$  in die Skip-Liste ein
- `Delete(key k)`: Entfernt einen Schlüssel  $k$  aus der Skip-Liste
- `Search(key k)`: Gibt `true` zurück, wenn der Schlüssel  $k$  in der Skip-Liste vorhanden ist, sonst `false`

Testen Sie Ihre Implementierung anhand verschiedener Zufallszahlenfolgen.