

## Übungen zur Vorlesung

**Algorithmen und Datenstrukturen**

WiSe 2025/26

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!
- > Musterlösungen werden bei Bedarf in den Übungen besprochen!

**Aufgabe 1:**

Viele Algorithmen benötigen eine Prioritätswarteschlange als Datenstruktur. Implementieren Sie eine Prioritätswarteschlange mit folgenden Operationen in C,C++,C#, Java oder Python:

1. `Insert(elem e, key x)`: Fügt ein Element  $e$  mit Schlüssel  $x$  ein
2. `GetMin()`: Liefert ein Element mit kleinstem Schlüssel
3. `ExtractMin()`: Liefert und entfernt ein Element mit kleinstem Schlüssel
4. `DecreaseKey(elem e, pos x, key k)`: Verringert (!) den Schlüssel eines Elements an Position  $x$  auf den Wert  $k$

Bestimmen Sie Zeit- und Platzkomplexität Ihrer implementierten Operationen!

**Aufgabe 2:**

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 3:**

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 4:**

Implementieren Sie Hashing mit offener Adressierung in C,C++, C#, Java oder Python anhand der Hashfunktion  $\tilde{h}(s) = s$  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.