

Auszuarbeiten bis 30.06.21

1. Hashtabelle (7 Punkte)

Realisieren Sie eine eigene generische Hashtabelle, die auf Wunsch der Verwenderin die Lineare Kollisionsstrategie, die Quadratische Kollisionsstrategie und Doppelhashing unterstützt. Vermeiden Sie bei Ihrem Design Codeverdopplung und andere „unschöne“ Codierungsvarianten.

2. Zusammenfügen 2er sortierter, einfach-verketteter Listen zu einer neuen sortierten Liste (9 Punkte)

Entwickeln Sie eine Methode `Merge`, die zwei einfach-verkettete, aufsteigend-sortierte Listen L_1 und L_2 des Datentyps `List` zu einer neuen, aufsteigend-sortierten Liste L_3 zusammenfügt.

Die beiden Listen werden als Eingabeparameter an die Funktion übergeben. Die neue Liste wird als Rückgabewert zurückgegeben.

Achten Sie bei der Realisierung der Methode `Merge` auch auf Effizienz und Robustheit! Bestehende Knoten sollen wiederverwendet werden!

3. Laufzeitkomplexität (4 Punkte)

a) Was beschreibt die O -Notation? Welche Bedingung ist für die asymptotische Laufzeitkomplexität von essentieller Bedeutung?

b) Erklären Sie die nachfolgende mathematische Definition der O -Notation

$$g(n) = O(f(n))$$

$$O(f(n)) = \left\{ \begin{array}{l} g(n) : \exists c, n_0 > 0 \\ \exists 0 \leq g(n) \leq c f(n), \forall n \geq n_0 \end{array} \right\}$$

$$g(n) = O(f(n)) \Leftrightarrow \lim_{n \rightarrow \infty} \left| \frac{g(n)}{f(n)} \right| \leq c$$