

Hausaufgabenblatt 5

Die Abgabe der Lösungen muss bis zum 13.07.21 um 23:59 Uhr erfolgen. Lösungen müssen per Mail in einer PDF-Datei (Dateiname „blatt_[nr]_[name1]_[name2]_[grpnr].pdf“) an die/den jeweilige*n Tutor*in geschickt werden. Email-Adressen sind unter <https://www.ibr.cs.tu-bs.de/alg/index.html> zu finden.

Hausaufgabe 1 (Hashing):

(1+1+2+4 Punkte)

Betrachte zunächst eine Hashfunktion $h(x)$ mit $\text{Prob}(h(x) = j) = \frac{1}{m}$, wobei m die Größe der Hashtabelle ist.

- Wie hoch ist die Wahrscheinlichkeit, dass das k -te Element kollisionsfrei eingefügt werden kann, wenn bereits $k - 1$ Zellen der Tabelle belegt sind?
- Wie hoch ist die Wahrscheinlichkeit, dass 3 Datensätze kollisionsfrei in eine Hashtabelle der Größe $m = 20$ eingefügt werden können? Wie hoch ist die Wahrscheinlichkeit bei 5 Datensätzen? Gib die Wahrscheinlichkeiten in Prozent an.

Anstatt Objekte in eine Liste zu schreiben, gibt es die Möglichkeit, Elementen ein anderes freies Feld in der Hashtabelle zuzuweisen, falls eine Kollision auftritt. Man nennt dieses Hashverfahren auch *offenes Hashing*. Dazu betrachtet man für eine Hashtabelle der Größe m zum Beispiel eine Hashfunktion der folgenden Form:

$$t(x, i) := h_1(x) + i \cdot h_2(x) \pmod{m}$$

Dabei sind $h_1 : \mathbb{N} \rightarrow \{0, \dots, m - 1\}$ und $h_2 : \mathbb{N} \rightarrow \{1, \dots, m - 1\}$ zwei Hashfunktionen. Der Wert i gibt dabei die Anzahl der Versuche an, ein Element x in die Hashtabelle einzufügen. Im Allgemeinen startet man mit $i = 0$.

- Warum dürfen Elemente mit diesem Hashverfahren nicht einfach aus der Tabelle gelöscht werden? Wie kann man das Problem umgehen?
- Betrachte ein anfangs leeres Array A der Größe 11, es gibt also die Speicherzellen $A[0], \dots, A[10]$. In diesem Array führen wir offenes Hashing mit der folgenden Hashfunktion $t(x, i)$ durch.

$$t(i, x) := 5x^2 - 4x + i \cdot h_2(x) \pmod{11} \quad \text{mit } h_2(x) := (2x + 3 \pmod{10}) + 1.$$

Dabei ist x ein einzusetzender Schlüssel und i die Nummer des Versuches, x in eine unbesetzte Speicherzelle des Arrays zu schreiben, beginnend bei $i = 0$. Berechne zu jedem der folgenden Schlüssel die Position, die er in A bekommt:

$$7, \quad 3, \quad 5, \quad 12, \quad 14, \quad 2$$

Dabei sollen die Schlüssel in der gegebenen Reihenfolge eingefügt werden und der Rechenweg soll klar erkennbar sein.

Hausaufgabe 2 (Reduktion):**(3+1+6+2 Punkte)**

In dieser Aufgabe betrachten wir PARTITION und wollen analysieren, wie schwer das Problem ist. Aus der Vorlesung ist bekannt, dass SUBSET SUM NP-vollständig ist.

- a) Reduziere PARTITION auf SUBSET SUM.
- b) Was sagt die Reduktion aus a) über die Komplexitätsklasse von PARTITION aus?
- c) Zeige: PARTITION ist NP-vollständig. (Hinweis: Führe dazu eine Reduktion von SUBSET SUM auf PARTITION durch.)
- d) Welche Folge(n) hat es, falls es einen Polynomialzeit-Algorithmus für PARTITION gibt? – Dann gibt es einen Polynomialzeit-Algorithmus für...

...jedes Problem in NP.

...jedes NP-vollständige Problem.

...jedes NP-schwere Problem.

Kreuze die richtige(n) Antwort(en) an und begründe deine Wahl.