

Algorithmen und Datenstrukturen  
SS 2019

Übungsblatt 9: Suchen II

Aufgabe 9-1 Bitvektoren

Hinweis: Man kann Bitvektoren auf zwei gegensätzliche Arten definieren. Dabei gibt es keine richtige Definition. Um es etwas intuitiver und Übung sowie Vorlesung konform zu machen, definieren wir: Ein Element ist genau dann in der Menge enthalten, wenn das entsprechende Bit den Wert 1 hat. Wir haben dies entsprechend auf den Vorlesungsfolien geändert.

Gegeben ist wieder die Sammlung von Informatik-Fachbüchern:

Nummer	Titel	Autor	Jahr	Kürzel
0	Design Patterns	E. Gamma, et al.	1994	DP
1	Clean Code	Robert C. Martin	2008	CC
2	Make Your Own Neural Network	Tariq Rashid	2016	MNN
3	Agile Software Development	Robert C. Martin	2002	AgSD
4	Introduction to Algorithms	T. H. Cormen, et al.	1989	IA
5	Functional Thinking: Paradigm Over Syntax	Neal Ford	2014	FuT
6	Extreme Programming Explained: Embrace Change	Kent Beck	1999	ExPE
7	Algorithms for Reinforcement Learning	Csaba Szepesvari	2010	AIRL
8	The Software Craftsman	Robert C. Martin	2014	TheSC
9	Test Driven Development: By Example	Kent Beck	2002	TDD
10	Programming Pearls	Jon Bentley	1986	PP
11	Building Your Own Compiler with C++	Jim Holmes	1994	BYOC

Das Universum  $U$  soll im folgendem alle obigen Bücher enthalten. Ein Eintrag  $\text{Bit}[i]$  eines Bitvektors steht dafür, ob ein Buch mit Nummer  $i$  in einer gegebenen Menge enthalten ist.

Hinweis: Sie dürfen die Vektoren auch als Zeilenvektoren schreiben.

- Geben Sie die Größe des Universums  $N$  an.
- Geben Sie den Bitvektor an, der das Universum repräsentiert. Wie viele Elemente (Bücher) enthält die Menge, die er repräsentiert.
- Welcher Bitvektor repräsentiert folgende Menge an Büchern  $M_c = \{\text{AgSD}, \text{TDD}, \text{BYOC}, \text{IA}, \text{TheSC}\}$ ?
- Welcher Menge repräsentiert der Bitvektor  $\text{Bit}(M_d) = (1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0)^T$ ?
- Wie könnte ein effizienter Algorithmus aussehen, der mittels zweier Bitvektoren  $\text{Bit}(M_1)$  und  $\text{Bit}(M_2)$  die Schnittmenge und Vereinigung der Mengen, die sie repräsentieren ( $M_1$  und  $M_2$ ), berechnet und diese wieder als Bitvektor  $\text{Bit}(M_{\text{result}})$  ausgibt? Geben Sie diesen Algorithmus möglichst formal korrekt in Pseudocode an. Wie groß ist die Laufzeit in O-Notation?

**Aufgabe 9-2**     *Hashing*

Gegeben sei folgende Hashfunktion  $h(k) = k \bmod 7$ . Fügen sie nun nacheinander folgende Schlüssel in eine Hashtabelle ein. Verwenden Sie dabei *offenes Hashing*:

- (a) 1, 7, 13, 15, 14, 5
- (b) c, z, a, j, v, k, l (*überlegen Sie Möglichkeiten, wie Sie Buchstaben mit der Hashfunktion  $h(k)$  in die Tabelle einfügen können*)

**Aufgabe 9-3**     *Geschlossenes und doppeltes Hashing*

Gegeben sind die Zahlen  $\{13, 7, 31, 19, 27, 42, 69, 96\}$  welche in eine Hashtabelle der Größe 11 eingeordnet werden soll.

- (a) Benutzen Sie geschlossenes Hashing mit  $h(k) = k \bmod 11$  und ordnen Sie die Zahlen ein. Kennzeichnen Sie Kollisionen und lösen Sie diese durch lineares Sondieren mit  $c_1 = 1$ .
- (b) Nutzen Sie nun Doppel-Hashing mit  $h(k) = k \bmod 11$ , der Sekundären Hashfunktion  $h'(k) = k \bmod 9 + 1$  und der Sondierungsfunktion  $h_j(k) = (h(k) + j * h'(k)) \bmod 11$ . Dokumentieren Sie Kollisionen und deren Lösung.