

Übungsblatt 3

Abgabe am 05.11.2018

Aufgabe 1: Stacks und Queues (3+4 Punkte)

Angenommen Sie möchten eine Queue-Datenstruktur mit den üblichen Funktionen $\text{ENQUEUE}(\cdot)$ und $\text{DEQUEUE}(\cdot)$ bereitstellen, wozu Ihnen “intern” allerdings nur *genau 2 Stacks* zur Verfügung stehen (also keine Arrays, Listen, oder anderes).

- Beschreiben Sie eine Implementierung der Queue (in Worten und in Pseudocode), die nur die beiden Stacks benutzt. Was können Sie über die worst-case-Laufzeit für eine $\text{ENQUEUE}(\cdot)$ - bzw. $\text{DEQUEUE}(\cdot)$ -Operation sagen?
- Nun wollten wir eine Methode betrachten, mit der man die Laufzeit mehrerer Operationen gemeinsam betrachtet:
 - Betrachten Sie eine beliebige Folge von insgesamt n_E $\text{ENQUEUE}(\cdot)$ und insgesamt n_D $\text{DEQUEUE}(\cdot)$ -Operationen mit $n_E + n_D = n$.
 - Berechnen Sie die worst-case-Laufzeit T_n für diese Folge von Operationen in Abhängigkeit von der Anzahl der Operationen n .
 - Dann heißt T_n/n die *amortisierte Laufzeit* dieser Operationen.

Falls nicht bereits geschehen, finden Sie eine Implementierung einer Queue, die wie oben nur zwei Stacks benutzt und amortisierte Laufzeit $\mathcal{O}(1)$ hat. Begründen Sie Ihre Antwort. Tipp: Betrachten Sie, welche Stack-Operationen ein Element v im schlimmsten Fall mit $\text{ENQUEUE}(\cdot)$ - bzw. $\text{DEQUEUE}(\cdot)$ durchläuft.

Aufgabe 2: k -närer Heap (2+2+2+1+2 Punkte)

In dieser Aufgabe benötige die HEAPIFY -Operation an der Wurzel eines Array-basierten k -nären Heaps mit n Elementen im worst-case $\lceil k \log_k(n) \rceil$ Schritte. Dabei sind auf jedem seiner $\lceil \log_k(n) \rceil$ vielen Level zusätzlich k Schritte zum Finden (und ggf. Vertauschen) des Maximums der k Kinder des aktuellen Knotens bereits berücksichtigt. Welche Wahl von $k \in \mathbb{N}$ minimiert diesen Ausdruck? Diese Frage wollen wir in dieser Aufgabe beantworten.

- Bestimmen Sie zunächst das Minimum der Funktion $f(x) = x \log_x(n)$ auf dem Definitionsbereich $\mathbb{R}_{>1}$ für festes, aber beliebiges n . Tipp: Kurvendiskussion — schreiben Sie dazu mittels Basiswechsel $x \log_x(n) = x \cdot \log_e(n) / \log_e(x)$.
- Nutzen Sie einen Plot von $f(x)$ aus (a) oder die Einsichten aus (a), um die beste Wahl $k^* \in \mathbb{N}$ zu ermitteln. Wie viele Schritte benötigt diese im worst-case bei Heap-Größe $n = 10^\ell$ für $\ell \in \{1, \dots, 9\}$? Vergleichen Sie dies mit dem Fall $k = 2$.
- Wie wirkt es sich auf die Gesamtlaufzeit von HEAPIFY aus, wenn jeder Knoten seine (bis zu k vielen) Kinder in einem separaten binären Max-Heap verwaltet? Bestimmen Sie hierfür insbesondere die notwendige Anzahl Schritte zum Finden (und. ggf. Austauschen) des Maximums der k Kinder des aktuellen Knotens.
- Führen Sie auf dem binären Heap $B = \begin{bmatrix} 9 & 8 & 6 & 7 & 3 & 5 & 2 \end{bmatrix}$ und dem ternären Heap $T = \begin{bmatrix} 9 & 7 & 5 & 8 & 3 & 6 & 2 \end{bmatrix}$ jeweils die Operation $\text{DECREASE}(9 \mapsto 1)$ aus. Wie viele Vertauschungen wurden jeweils durchgeführt?
- Wir bezeichnen ein Array A mit paarweise unterschiedlichen Einträgen als $2/3$ -kompatibel, wenn sowohl der jeweils per Level-Order definierte binäre Heap H_2 als auch der ternäre Heap H_3 beide die Max-Heap-Eigenschaft erfüllen. Überdenken Sie folgende Aussage: “Für alle $2/3$ -kompatiblen Arrays gilt: In H_3 benötigt keine DECREASE -Operation auf der Wurzel mehr Vertauschungen als dieselbe DECREASE -Operation auf der Wurzel in H_2 ”. Stimmt das? Beweisen oder widerlegen Sie!

Aufgabe 3: Hashing (1 + 1 + 1 + 1 Punkte)

Sei $\mathbb{N} := \{0, 1, 2, \dots\}$. Wir schreiben $a\mathbb{N} + b := \{a \cdot t + b \mid t \in \mathbb{N}\}$ für $a, b \in \mathbb{N}$, beispielsweise $5\mathbb{N} + 2 = \{2, 7, 12, 17, \dots\}$. Geben Sie in dieser Schreibweise jeweils die Menge aller Keys k an (inkl. Begründung), die auf der letzten Position (Index 10) in einer Hashtabelle der Größe 11 kollidieren, wenn folgende Hashfunktionen verwendet werden:

(a) $h(k) = k \mod 11$

(c) $h(k) = (k^2 + 10) \mod 11$

(b) $h(k) = 2k \mod 11$

(d) $h(k) = (3^k - 1) \mod 11$

Tipp: Für (c) und (d) rufen Sie sich die Bedeutung der Primfaktorzerlegung einer natürlichen Zahl in Erinnerung, insbesondere wenn sie quadriert wird.