
Übungsblatt 3

Ausgabe: 09.05.2018 – 15:30
Abgabe: 16.05.2018 – 13:00

A Amortisierte Analyse (3 Punkte)

Auf einer Datenstruktur wird eine Sequenz $\sigma = (\sigma_1, \dots, \sigma_n)$ von Operationen ausgeführt ($n = 2^m$ für ein $m \in \mathbb{N}$). Die Operation σ_i kostet i , wenn i eine Potenz von zwei ist, sonst 2.

- Berechnen Sie die Kosten für 16 Operationen $T(16)$.
- Geben Sie eine geschlossene Form $T(n)$ für die Kosten von n Operationen an.
- Wie groß sind die amortisierten Kosten pro Operation? Schätzen Sie die amortisierten Kosten auch in \mathcal{O} -Notation ab.

B XOR-Listen

In der Vorlesung wurden doppelt verkettete Listen L mit zwei Zeigern pro Listenelement x , $x.\text{prev}$ und $x.\text{next}$, implementiert. Nehmen Sie an, dass alle Zeigerwerte als k -bit Integer-Zahlen interpretiert werden können. Nehmen Sie weiter für die Aufgabenstellung an, dass jedes Listenelement x ein Feld $x.\text{key}$ hat, nach dem gesucht werden kann.

B.1 Grundlegendes (2 Punkte)

Beweisen Sie für zwei k -Bit-Integer I und J und den bitweisen XOR-Operator \oplus die Gültigkeit von:

$$(I \oplus J) \oplus J = I .$$

Sie können also zu gegebenem $K = (I \oplus J)$ und I das passende J wiederfinden. Beschreiben Sie, wie eine doppelt verkettete Liste mit nur einem Zeiger $x.\text{np}$ pro Listenelement x damit implementiert werden kann, so dass die Funktionen aus Teilaufgabe B.2 effizient implementiert werden können.

Tipp: Lassen sich die Zeiger auf zwei (wie ausgewählte?) Listenelemente als I und J verwenden? Für Anfang und Ende der XOR-Liste bieten sich "Dummy"-Elemente an.

B.2 Pseudocode (2 Punkte)

Implementieren Sie die beiden Funktionen aus der Vorlesung `findNext` in $\mathcal{O}(|L|)$ und `pushFront` in $\mathcal{O}(1)$ für XOR-Listen. Implementieren Sie außerdem eine Methode `invert`, welche die XOR-Liste in $\mathcal{O}(1)$ spiegelt.

C Doppelt-verkettete Listen (3 Punkte)

Sie sollen eine unbeschränkte Queue (Warteschlange) mittels einer doppelt verketteten Liste konstruieren. Geben Sie hierzu den Pseudocode zur Beschreibung eines Listenelements x sowie der Operationen `pushBack` und

`popFront` an, so dass diese jeweils nur konstante Zeit benötigen. Verwenden Sie nur elementare Operationen, insbesondere *keine* Subroutinen aus der Vorlesung (z.B. `splice`).

Deckblatt Übungsblatt 3

Tutoriumsnummer:

Name	Matrikelnummer	Unterschrift

Mit unseren Unterschriften bestätigen wir, dass wir die Aufgaben eigenständig gelöst haben.

Hinweis: Das Übungsblatt darf in Gruppen von bis zu zwei Personen bearbeitet werden. Beide Personen müssen demselben Tutorium zugeteilt sein. Möchte jemand seine Abgaben-Gruppe innerhalb des Semesters wechseln, so ist dies im Voraus mit dem Tutor abzusprechen. **Bitte tragen Sie oben groß die Nummer Ihres Tutoriums ein.** Die Lösung des Übungsblattes ist in jedem Fall mit diesem Deckblatt abzugeben.

Bewertung (durch Tutor):