

10. Übungsblatt

Aufgabe 1: Dynamische Tabellen

4 Punkte

Eine alternative Implementierung für dynamische Tabellen verringert die Größe der Tabelle auf $2/3$ der Ursprungsgröße, falls der Lastfaktor unter $1/3$ fällt. Zeigen Sie, dass die amortisierten Kosten für das Entfernen auch in dieser Implementierung durch eine Konstante nach oben beschränkt sind. Verwenden Sie als Potentialfunktion

$$\Phi(T) := |2k - s|.$$

Hierbei sei T die Tabelle, s die Größe der Tabelle und k die Anzahl der enthaltenen Elemente. Als Lastfaktor einer Tabelle bezeichnen wir den Wert k/s .

Aufgabe 2: Vorrangwarteschlangen

3 Punkte

Beweisen Sie, dass es nicht möglich ist, eine Vorrangwarteschlange zu implementieren, in der alle Operationen amortisiert konstante Laufzeit haben, falls die einzige Operation auf Schlüsseln deren Vergleich ist.

Aufgabe 3: Worst-Case Analyse

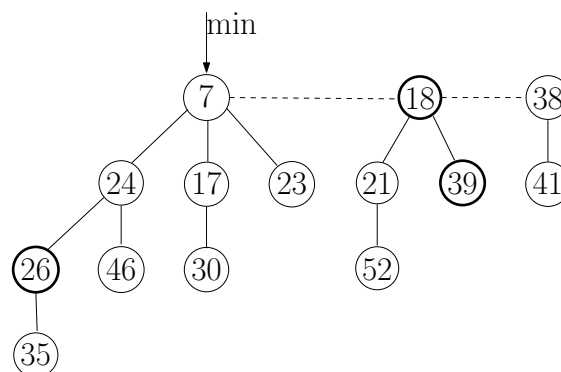
3 Punkte

Geben Sie in Θ -Notation die Worst-Case Laufzeit der Operation `deletemin` auf Fibonacci-Heaps an und begründen Sie Ihre Aussage.

Aufgabe 4: Fibonacci-Heaps

5 Punkte

Führen Sie auf nachstehendem Fibonacci-Heap die Operation `deletemin` und danach die Operationen `decreasekey(46, 12)` und `decreasekey(35, 17)` aus. Die fett umrandeten Knoten sollen hierbei markiert seien. Die Operation `decreasekey(n, m)` setzt den Schlüssel des Knotens mit Schlüssel n auf m herab. Stellen Sie Ihre Lösung derart vor, dass die einzelnen Schritte der Operation nachvollziehbar sind. Einfügeoperationen in die Wurzelliste sollen immer links vom Minimum-Zeiger stattfinden, der Konsolidieren-Schritt soll die Wurzelliste von links nach rechts durchlaufen.



Abgabe: Montag, 10. Juli 2006, 14 Uhr, in den entsprechenden Briefkästen in Gebäude 051. Die Übungsblätter können in Gruppen à maximal 2 Personen bearbeitet werden. Vermerken Sie die Namen und Matrikelnummern der an der Bearbeitung beteiligten Personen.

Beachten Sie bitte auch die aktuellen Hinweise unter

www.informatik.uni-freiburg.de/~ipr → Teaching → Informatik II