



8. Übungsblatt zu Grundzüge von Algorithmen und Datenstrukturen, WS 12/13

Prof. Dr. Markus Bläser, Radu Curticapean, Christian Engels
<http://www-cc.cs.uni-sb.de/course/38/>

Abgabe: Donnerstag, 10. Januar 2013, 12:00 Uhr

Aufgabe 8.1 Nehmen Sie an, wir benötigen eine Queue. Doch leider haben wir nur zwei Stacks zur Hand. Soll ein Element zur Queue hinzugefügt werden, so wird es auf den ersten Stack gepusht. Soll ein Element aus der Queue entfernt werden, so wird das oberste Element vom zweiten Stack gepopt, sofern dieser nicht leer ist. Ist der zweite Stack leer, so wird vorher der Inhalt des ersten Stacks auf den zweiten Stack kopiert mittels push und pop. Dabei dreht sich die Reihenfolge der Elemente um.

- Überlegen Sie sich, dass dies tatsächlich die Funktionalität einer Queue implementiert.
- Welche Worst-Case Laufzeiten haben Einfügen in und Entfernen aus der so implementierten Queue?
- Zeigen Sie mit der Aggregat-Methode, dass Einfügen in und Entfernen aus der Queue amortisierte konstante Zeit hat.
- Zeigen Sie dasselbe mit Hilfe von Potentialfunktionen. (Hinweis: Wählen Sie eine Funktion, die proportional zur Größe des ersten Stacks ist.)

Aufgabe 8.2 Führen Sie folgende Operationen auf einem anfangs leeren Fibonacci-Heap aus. Zeichnen Sie den Heap nach jedem Schritt. Einige Schritte im Skript sind nicht komplett eindeutig (wo genau wird in den Ring eingefügt, etc.), diese dürfen Sie beliebig ausgestalten.

- Fügen Sie die Schlüssel $1, 2, \dots, 9$ ein.
- Extrahieren Sie das Minimum.
- Sie sollten nun einen Binomialbaum mit acht Elementen haben. Dort gibt es genau einen Knoten der zwei Kinder hat. Setzen Sie nacheinander jeweils den Schlüssel dieser zwei Kinder auf 0.