technische universität dortmund

Fakultät für Informatik
Lehrstuhl 11 / Algorithm Engineering

Prof. Dr. Petra Mutzel
Carsten Gutwenger

Wintersemester 2008/09

Algorithmen und Datenstrukturen
Übungsblatt 1

Ausgabe: 24. Oktober — Besprechung: 4./6. November

Allgemeine Hinweise:

- Es sollen nicht alle Aufgaben von allen Teams bearbeitet werden. Welches Team welche Aufgabe präsentieren bzw. schriftlich bearbeiten soll, ist auf der Webseite zur Übung vermerkt!
- Wenn bei Literaturaufgaben ein Artikel als Grundlage angegeben wird, bedeutet das nicht, dass *nur* dieser Artikel gelesen werden soll. Versuchen Sie auch (wenn sinnvoll / möglich), nach weiteren Informationen zu recherchieren, insbesondere sind auch experimentelle Studien zu den jeweiligen Datenstrukturen für uns interessant.
- Präsentationen sollen etwa 20 Minuten lang sein (exklusive Fragen und Diskussion). Versuchen Sie bei ihrer Präsentation diese Zeit auch einzuhalten und beschränken Sie ihre Darstellung (wenn notwendig) auf das für das Verständnis Wesentliche, um diese Zeit optimal zu nutzen.

Aufgabe 1.1 (Fibonacci-Heaps)

In der Vorlesung wurde zur amortisierten Analyse von Fibonacci-Heaps die Potenzialmethode verwendet. Führen Sie stattdessen die amortisierte Analyse von Fibonacci-Heaps mit Hilfe der Buchhaltermethode (=Bankkontomethode) durch.

Aufgabe 1.2 (HoT-Queues)

Stellen Sie die Datenstruktur der *Heap-on-Top Priority-Queues* zur Realisierung von Prioritätswarteschlangen vor. Beschreiben Sie die generelle Funktionsweise der Datenstruktur und welche Laufzeiten für die Priority-Queue Operationen erreicht werden. Gehen Sie dabei auch darauf ein, wie diese Laufzeitgarantien gezeigt werden können (Beweisidee).

Grundlage ist folgender Artikel: B. V. Cherkassky, A. V. Goldberg, and C. Silverstein, *Buckets, heaps, lists, and monotone priority queues*, SIAM Journal on Computing, 28(4), pp. 1326–1346, 1999.

Aufgabe 1.3 (Redistributive Heaps)

Stellen Sie die Datenstruktur der *redistributiven Heaps* zur Realisierung von Prioritätswarteschlangen vor. Beschreiben Sie die generelle Funktionsweise der Datenstruktur und welche Laufzeiten für die Priority-Queue Operationen erreicht werden. Gehen Sie dabei auch darauf ein, wie diese Laufzeitgarantien gezeigt werden können (Beweisidee).

Grundlage ist folgender Artikel: R. K. Ahuja, K. Mehlhorn, J. B. Orlin, and R. E. Tarjan, *Faster algorithms for the shortest path problem*, Journal of the ACM, 37(2), pp. 213–223, 1990.

Aufgabe 1.4 (Externe Listen)

Beschreiben Sie eine effiziente Implementierung für externe lineare Listen (d.h. geordnete dynamische Listen) und analysieren Sie deren I/O-Verhalten. Die lineare Liste soll folgende Operationen unterstützen:

- Search(x): Suche Element x in der Liste.
- Insert(x): Füge Element x an der bzgl. der linearen Ordnung der Elemente korrekten Stelle in die Liste ein.
- Delete(x): Lösche Element x aus der Liste.