Christian-Albrechts-Universität zu Kiel
Institut für Informatik und Praktische Mathematik

Prof. Dr. W.-P. de Roever
Martin Steffen

Algorithmen & Datenstrukturen

Wintersemester 2002/2003 | **Serie 2** | 29. Oktober 2002

Thema : Sortieren

Ausgabetermin: 29. Oktober 2002

Abgabe: 14. November 2002

Aufgabe 1 (Heaps und Prioritätswarteschlangen) Der Heap-Datentyp aus der Vorlesung ist nicht nur als Hilfsstruktur zum Sortieren geeignet.

- Eine Anwendung von Heaps sind Prioritätswarteschlangen (*priority queues*). Eine Prioritätswarteschlange ist ein Sammlung geordneter Elemente; das Einfügen darf in jeder beliebigen Reihenfolge geschehen, das Entfernen und der lesende Zugriff ist nur auf das Element mit der höchsten Priorität möglich. Diese Struktur kann man als *Menge* S von Elementen sehen, denen ein *Schlüssel* zugeordnet ist, wobei die Schlüssel geordnet sind. Folgende Operationen werden unterstützt:

  - $\mathit{Insert}(S, x)$: fügt x der Menge S hinzu
  - $\mathit{Maximum}(S)$: gibt ein Element mit dem größten Schlüssel zurück
  - $\mathit{Extract\_Max}(S)$: gibt ein Element mit dem größten Schlüssel zurück und entfernt dieses.

  Implementieren Sie eine Prioritätswarteschlange mit Hilfe der Heap-Datenstruktur.

- Eine Operation, die im allgemeinen *nicht* von Heaps (oder Prioritätswarteschlangen) verlangt wird, ist das *Verschmelzen* ($\mathit{Merge}(S_1, S_2)$) zweier Heaps in einen neuen. Implentieren Sie eine solche Prozedur. Was ist die *worst-case*-Komplexität Ihrer Lösung in Abhängigkeit von der Knotenanzahl?