

Übung 1:

Ausgabetermin: 22. April 1999

Abgabe: 28. April 1999

Aufgabe 1: [Listengenerator]

Zum empirischen Vergleich der Laufzeiten verschiedener Sortieralgorithmen sind *zufällig erzeugte Listen* von Zahlen nützlich. Schreiben Sie eine Funktion *random_list*, die eine Sequenz von natürlichen Zahlen erzeugt. Die Länge der Sequenz und der Bereich der erzeugten Zahlen sei Parameter der Funktion. Verwenden Sie diese Funktion um die mittlere Komplexität von *Heap-Sort* abzuschätzen

Aufgabe 2: [Prioritätswarteschlange]

Der Heap-Datentyp aus der Vorlesung ist nicht nur als Hilfsstruktur zum Sortieren geeignet. Eine Anwendung von Heaps sind Prioritätswarteschlangen (*priority queues*). Diese Struktur kann man als *Menge S* von Elementen sehen, denen ein *Schlüssel* zugeordnet ist.

- *Insert(S, x)*: fügt x der Menge S hinzu
- *Maximum(S)*: gibt ein Element mit dem größten Schlüssel zurück
- *Extract_Max(S)*: gibt ein Element mit dem größten Schlüssel zurück und entfernt dieses.

Implementieren Sie eine Prioritätswarteschlange mit Hilfe der Heap-Datenstruktur.