

## Algorithmen und Programmierung III

Abgabe 9.11.2012, 12 Uhr

---

### Aufgabe 1

7 Punkte

- (a) Implementieren Sie *Slowsort* in JAVA und bestimmen Sie experimentell seine Laufzeit für verschiedene Problemgrößen auf Ihrem Rechner. Beschreiben Sie, wie Sie systematisch alle Permutationen erzeugen.
- (b) Die “randomisierte Variante” von *Slowsort* besteht darin, eine zufällige Permutation auszuwählen, diese auf die Eingabefolge anzuwenden und zu prüfen, ob sie dann sortiert ist. Falls nicht wird das Experiment wiederholt. Implementieren Sie dieses Verfahren in JAVA und bestimmen Sie experimentell seine Laufzeit für verschiedene Problemgrößen auf Ihrem Rechner. Beschreiben Sie, wie Sie eine zufällige Permutation erzeugen, so dass jede gleichwahrscheinlich ist.

### Aufgabe 2

7 Punkte

- (a) Geben Sie einen einfachen Algorithmus zum Bestimmen des Minimums und des Maximums einer Folge von  $n$  vergleichbaren Elementen an und analysieren Sie die erforderliche (genaue) Anzahl von Vergleichen als Funktion von  $n$ .
- (b) Lösen Sie das gleiche Problem mit Teile-und-herrsche, indem Sie Folgen der Länge größer zwei in zwei Hälften zerlegen. Sie dürfen dabei annehmen, dass  $n$  eine Zweierpotenz ist.

### Aufgabe 3

6 Punkte

Nehmen Sie an, Sie hätten eine Sammlung von  $n$  Schrauben und  $n$  Schraubenmutter, wo jede Schraube in genau eine Mutter passt. Rein äußerlich ist kein Unterschied festzustellen, nur durch ausprobieren einer Schraube an einer Mutter kann man feststellen ob die Schraube zu dick ist, zu dünn ist oder passt. Entwerfen Sie einen möglichst effizienten Algorithmus, der zu jeder Schraube die passende Mutter findet und analysieren Sie die Anzahl der Tests, die dafür notwendig sind. Für letzteres dürfen Analysen aus der Vorlesung benutzt werden.