

## 6. Übungsblatt zu Algorithmen I im SS 2015

<https://crypto.itl.kit.edu/algo-rose15>  
{staudt,striecks}@kit.edu

### Aufgabe 1 (*Algorithmen-Design*, 2 + 2 Punkte)

Gegeben sei eine sortierte Liste mit  $n$  Elementen. Nun haben Sie  $k \leq n$  neue unsortierte Elemente in einem Array, die in die sortierte Liste eingefügt werden sollen, sodass die Liste nach dem Einfügen weiter sortiert ist.

- Geben Sie einen Algorithmus an, der Laufzeit  $\mathcal{O}(kn)$  hat und das Problem löst.
- Geben Sie nun einen Algorithmus an, der Laufzeit  $\mathcal{O}(k \log k + n)$  hat und das Problem löst.

### Aufgabe 2 (*Paralleles Sortieren*, 3 Punkte)

Die Ausführungszeit eines parallelen Algorithmus zum vergleichsbasierten Sortieren von  $n$  Elementen auf  $p$  Prozessoren sei

$$T(p) \in \Theta\left(\frac{n \log^2 n}{p}\right).$$

Geben Sie den absoluten Speedup (Beschleunigung) und die absolute Effizienz an. Der Speedup ist das Verhältnis von sequenzieller Laufzeit zu paralleler Laufzeit in Abhängigkeit von  $p$ . Die Effizienz ist definiert als Speedup geteilt durch die Anzahl der Prozessoren, wieder in Abhängigkeit von  $p$ . Wie muss die Prozessorzahl mit der Eingabegröße asymptotisch steigen, damit der Speedup konstant bleibt? Nehmen Sie immer den schlechtesten Fall an und rechnen Sie im  $\Theta$ -Kalkül.

### Aufgabe 3 (*Doktor Meta*, 2 + 2 Punkte)

Der ebenso geniale wie fundamental fehlentworfene Robotergehilfe und Untersuperbösewicht Røbøt ist auf einem Bein und einem Rad auf dem Weg durch Doktor Metas geheime Basis in den Tunnelsystemen unter Karlsruhe. Seine neue Entdeckung sollte allen Verdacht beseitigen, dass ihm einige essentielle Schaltkreise fehlen — er hat eine Prioritätsliste entwickelt, die für beliebige Objekte und Ordnungen funktioniert und dennoch alle Prioritätslisten-Operationen (insert, deleteMin) in  $\mathcal{O}(1)$  erledigen kann. Doktor Meta ist sehr interessiert an dieser Datenstruktur. Røbøt kann sich gerade nicht an den Quellcode erinnern, aber Doktor Meta vermutet, dass er in der Zwischenzeit schon einen Algorithmus auf Basis der neuen Prioritätsliste entwickeln kann, der seine stetig wachsende Liste von Gegenspielern nach ihrer Bedrohung sortiert.

- Sei die Prioritätslisten-Datenstruktur RøbøList gegeben, welche die Operationen insert und deleteMin (wie in der Vorlesung) abstrakt bereithält. Nehmen Sie an, dass beide Operationen in konstanter Zeit ausgeführt werden können. Nutzen Sie RøbøList, um einen Algorithmus anzugeben, der ein Array  $A[1..n]$ , für  $n \in \mathbb{N}$ , von natürlichen Zahlen in  $\mathcal{O}(n)$  sortiert. Geben Sie den Algorithmus in Pseudocode an.
- Nehmen Sie an, dass eine RøbøList-Implementierung der deleteMin-Operation ausschließlich vergleichsbasiert und deterministisch arbeitet. Begründen Sie ausführlich, warum die Laufzeit einer derartigen deleteMin-Implementierung im schlechtesten Fall eine untere Schranke von  $\Omega(\log n)$  besitzt.

**Ausgabe:** Mittwoch, 20.5.2014

**Abgabe:** Freitag, 29.5.2014, 12:45 im Briefkasten im Untergeschoss von Gebäude 50.34