

## 5. Übungsblatt zu Algorithmen I im SS 2015

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

### Aufgabe 1 (Kombinatorik, 1 + 2 + 1 Punkte)

Aus der Vorlesung Grundbegriffe der Informatik wissen Sie, dass ein Wort  $w$  über einem Alphabet  $\Sigma$  eine Folge von Zeichen aus  $\Sigma$  ist.

- Sei  $\Sigma = \{x, y, z\}$  gegeben. Betrachten Sie das Wort  $w_1 = zxyx \in \Sigma^4$ . Wie viele verschiedene Wörter können durch Umstellen der Zeichen von  $w_1$  konstruiert werden? Geben Sie alle Wörter an.
- Sei  $\Sigma = \{1, \dots, j\}$ , für  $j \in \mathbb{N}$ , gegeben. Betrachten Sie das Wort  $w_2 \in \Sigma^n$ , für  $n \in \mathbb{N}_0$ . Wie viele verschiedene Wörter können durch Umstellen der Zeichen von  $w_2$  konstruiert werden? Begründen Sie Ihre Antwort. **Hinweis:**  $0! = 1$ .
- Sei  $\Sigma = \{A, e, g, h, i, l, m, n, o, r, t\}$  gegeben. Betrachten Sie die Wörter  $w_3 = \text{Algorithmen} \in \Sigma^{11}$  und  $w_4 = \text{1AghhbAArAA} \in \Sigma^{11}$ . Wie viele verschiedene Wörter können durch Umstellen der Zeichen von  $w_3$  und  $w_4$  jeweils konstruiert werden? Tipp: Versuchen Sie nicht, alle anzugeben. Nutzen Sie das Ergebnis aus b).

### Aufgabe 2 (Sortieren, 1 + 1 + 2 + 2 Punkte)

Sortieren Sie die Ziffern Ihrer Matrikelnummer mittels den in den Teilaufgaben angegebenen Algorithmen aus der Vorlesung. (Falls Sie zu zweit abgeben, genügt eine von beiden.)

- Bestimmen Sie die Anzahl von Inversionen in Ihrer Matrikelnummer.
- Benutzen Sie Insertionsort. Geben Sie den Zustand des Feldes nach jedem Einfüge-Schritt an.
- Benutzen Sie Mergesort. Verwenden Sie das Schema aus dem Beispiel auf der Vorlesung-Folie 173.
- Benutzen Sie Quicksort. Verwenden Sie das Schema aus dem Rekursions-Beispiel auf der Vorlesung-Folie 197. Als Pivot soll das erste Element verwendet werden.

### Aufgabe 3 (Sortieralgorithmen sortieren – Doktor Meta lässt grüßen, 2 Punkte)

Ordnen Sie die folgenden fünf Sortieralgorithmen aufsteigend nach Average-case-Komplexität: Heap-Sort, Merge-Sort, Quick-Sort, Insertion-Sort und Bucket-Sort für Schlüsselwerte in  $\mathcal{O}(n)$ . Annotieren Sie jeweils die Average-case-Komplexität.

### Aufgabe 4 (In-place-Duplikaterkennung, 4 Punkte)

Gegeben ist ein Array  $A : \text{Array}[0..n-1]$  der Größe  $n$ . Es enthält Zahlen aus  $\{0, \dots, n-1\}$ . Geben Sie einen *In-place*-Algorithmus an, der eine Zahl ausgibt, die doppelt vorkommt oder ausgibt, dass alle Zahlen verschieden sind. Die Laufzeit Ihres Algorithmus darf  $\mathcal{O}(n)$  nicht übersteigen. Geben Sie auch die Best-Case-Laufzeit Ihres Algorithmus an. **Hinweis:** Es ist erlaubt, das ursprüngliche Array zu verändern. Für einen *In-place*-Algorithmus steht nur  $\mathcal{O}(1)$  zusätzlicher Speicher zur Verfügung.

**Ausgabe:** Mittwoch, 13.5.2014

**Abgabe:** Freitag, 22.5.2014, 12.45 Uhr im Briefkasten im Untergeschoss von Gebäude 50.34.