



Prof. Dr. Sebastian Wild  
Dr. Nikolaus Glombiewski

Übungen zur Vorlesung  
**Effiziente Algorithmen**

Abgabe: 07.02.2024,  
bis **spätestens** 19:00 Uhr  
über die ILIAS Plattform

## Übungsblatt 12

### Aufgabe 12.1: Algorithmen Entwurf

(6 Punkte)

Betrachten Sie folgendes Spiel:

Gegeben seien eine gerade Anzahl  $n$  Münzen, welche auf einer Linie platziert werden. Münzen können unterschiedliche Werte haben. Zwei Spieler entfernen abwechselnd eine Münze von einem beliebigen Ende der Linie. Derjenige Spieler, der am Ende des Spiels den größten Gesamtwert erhalten hat, gewinnt.

Beispiel: Es liegen Münze mit folgenden Werten in folgender Linie aus:

6, 5, 2, 7, 3, 5

Spieler 1 nimmt zunächst die 6 vom linken Ende. Spieler B nimmt Münze 5 vom rechten Ende. Spieler 2 die Münze 5 vom linken Ende. Spieler 2 dann Münze 3 vom rechten Ende. Spieler 1 nimmt die Münze 7 vom linken Ende. Spieler 1 bleibt nur noch die Münze 2.

Spieler 1 hat einen Wert von 18 erspielt. Spieler 2 einen Wert von 10. Damit hat Spieler 1 gewonnen.

- Zeigen oder widerlegen Sie, dass folgende Greedy Strategie eine optimale Lösung liefert: Wähle immer den größten verfügbaren Wert.
- Entwickeln Sie für Spieler 1, unter der Annahme dass Spieler 2 immer optimal spielt, eine Lösung auf Basis von dynamischen Programmieren.

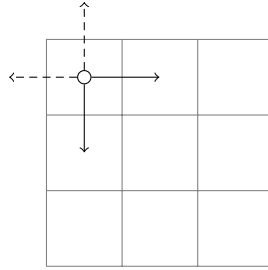
### Aufgabe 12.2: Dynamisches Programmieren

(6 Punkte)

Sie sind auf einer einsamen Insel in der Form einer  $n \times m$  Matrix  $M$  gestrandet.

Aktuell befinden Sie sich an Position  $(i, j) \in \{0, \dots, n-1\} \times \{0, \dots, m-1\}$  der Insel und Sie können einen *Schritt* nach oben, unten, links oder rechts machen um sich ein Feld weiterzubewegen. Da es hier Haie gibt, dürfen Sie dabei auf keinen Fall die Insel verlassen, also kein Feld  $(i', j') \notin \{0, \dots, n-1\} \times \{0, \dots, m-1\}$  betreten.

Ihre Fitnessapp sagt, dass Sie heute noch  $\ell \in \mathbb{N}$  Schritte gehen müssen, um Ihr Tagesziel zu erreichen. Es gilt nun, die *Anzahl* der verschiedenen Routen zu ermitteln, die Sie mit genau  $\ell$  Schritten gehen können, wenn Sie in Position  $(i, j)$  starten und die Insel nicht verlassen dürfen.



Geben Sie einen auf dynamischer Programmierung basierenden Algorithmus an, der die Aufgabe für eine gegebene  $n \times m$  Matrix, eine gegebene Anfangsposition  $(i, j)$  und eine gegebene Schrittzahl  $\ell$  in  $\mathcal{O}(n \cdot m \cdot \ell)$  Zeit und  $\mathcal{O}(n \cdot m \cdot \ell)$  Speicherbedarf löst. Wenden Sie hierbei den **Bottom-Up**-Ansatz an.

### Aufgabe 12.3: Indexierung von Texten

(8 Punkte)

Betrachten Sie den Text  $T = \text{abbabbaa\$}$ .

Was ist in diesem Fall  $n$ ? (Befolgen Sie die Konventionen aus der Vorlesung!).

Erstellen/zeichnen Sie folgende Strukturen

1. den standard (nicht komprimierten) Trie für alle Suffixe von  $T$ ,
2. dem Suffix Tree von  $T$ ,
3. das Suffix Array  $L[0..n]$  von  $T$ ,
4. das inverse Suffix Array  $R[0..n]$ , und
5. das LCP Array.

Lösen sie anschließend die folgenden Aufgaben:

- a) Beschreiben Sie, wie man den Suffix Tree aus den Arrays zu rekonstruieren kann (versuchen Sie hierbei sich zunächst auf  $L$  und  $LCP$  zu beschränken)
- b) Annotieren Sie die inneren Knoten des Suffix Trees mit der String Tiefe. Beschreiben Sie, wie String Tiefe und das LCP Array zusammenhängen.
- c) Verwenden Sie Ihre Strukturen, um den längsten, sich wiederholenden Substring in  $T$  zu bestimmen.