

---

## Algorithmen und Berechnungskomplexität I WS 15/16

Universität Bonn, Institut für Informatik, Abteilung I

5. Aufgabenblatt zur Vorlesung

Abgabe: 24.11. (12<sup>30</sup>)

---

### Aufgabe 17: Dynamische Programmierung (4 Punkte)

Seien  $n, k \in \mathbb{N}_0$  mit  $0 \leq k \leq n$  gegeben. Der Binomialkoeffizient  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$  kann auch unter Zuhilfenahme folgender Gleichung berechnet werden:

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

Entwickeln Sie einen Algorithmus, der diese Rechenvorschrift mit einem Speicherbedarf in  $O(k)$  mit Dynamischer Programmierung implementiert.

### Aufgabe 18: Greedy Algorithmen (4 Punkte)

Der Kaffeeautomat im Informatikgebäude bietet Heißgetränke zu verschiedenen Preisen an. Um nach einem Kauf Wechselgeld der Höhe  $x$  auszugeben, stehen dem Automat Münzen mit den Werten  $c_1, \dots, c_k$  zur Verfügung, wobei jeweils  $c_i < c_{i+1}$  gilt. In Euro z.B.: 0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1, 2. Die Anzahl an Münzen mit dem Wert  $c_i$ , die der Automat zu Beginn des Wechselvorgangs besitzt, ist mit  $n_i$  gegeben.

1. Beschreiben Sie einen Greedy-Algorithmus, der das Wechselgeld in Euro erstattet. Aus praktischen Gründen soll die Anzahl zurückgegebener Münzen so klein wie möglich sein.
2. Geben Sie ein Beispiel für eine (fiktive) Währung an, bei welcher der Greedy-Algorithmus kein optimales Ergebnis liefert. Nennen Sie eine Bedingung, die an eine Währung mit  $k \geq 3$  unterschiedlichen Münzwerten gestellt werden kann, sodass das Greedy Verfahren immer ein optimales Ergebnis liefert! Begründen Sie, warum Ihre Bedingung dies garantiert.

*Bitte wenden!*

