

ÜBUNGSSERIE 2

Algorithmen & Datenstrukturen AD1 / FS 2019

AD1 Team

Aufgabe 1 (Wiederholung Folgen)

Bestimmen Sie das n -te Glied (a_n) der Folge (wieder in *rekursiv*, *iterativ* und *explizit* Form (als Polynom)):

1,5,9,13,17,...

Aufgabe 2 (Wiederholung Reihen)

Bestimmen Sie entsprechend wieder die Summenformel (s_n) der Folge aus Aufgabe 1.

Aufgabe 3 (Konzepte)

Heutzutage sind die meisten Warenautomaten in der Lage Wechselgeld auszugeben. Die einfachste Möglichkeit wäre das Restgeld in Fünf-Rappen Stücken auszugeben. Das wäre das erste Mal ganz lustig, aber allerhöchstens das erste Mal.

Eine bessere Methode versucht den Betrag mit einer möglichst **minimalen Münzzahl** zu bestimmen. Dieser (Greedy-) Algorithmus hat die allgemeine Form:

- 1) wähle die grösst-mögliche Münze aus dem Zielwert aus,
- 2) ziehe den Wert vom Zielwert ab,
- 3) gehe zu Schritt 1 bis Differenz von Münzwert und Zielwert gleich Null.

Beispiel: 28 Franken lassen sich zusammensetzen aus

- 1) einer 20-er Note
- 2) einem 5, einem 2 und einem 1 Franken Stück

(a) Implementieren Sie diesen Algorithmus

(Unterstützte Werte: Münzen 1,2,5 / Noten 10,20,50,100,200).

(b) Auch dieser Algorithmus ist nicht optimal, da er davon ausgeht, dass sämtliche Geldstücke unbeschränkt oft vorliegen. Geben Sie eine Situation an, in welcher der Algorithmus versagt, obwohl es theoretisch möglich wäre den Betrag exakt auszugeben.

Aufgabe 4 (O-Notation)

Analysieren Sie die Laufzeit des folgenden Programmfragments:

```
for(int i=0;i<n;i++)
{
    for(int j=0;j<10_000_000;j++)
    {
        s
    }
}
```

mit $T_s(n) \in O(1)$.

Geben Sie die Laufzeit in expliziter Form und in der O-Notation an.

Aufgabe 5 (O-Notation)

Analysieren Sie die Laufzeit des folgenden Programmfragments:

```
for(int i=0;i<n;i++)
{
    for(int j=i;j<=n;j++)
    {
        s
    }
}
```

mit $T_s(n) \in O(1)$.

Geben Sie die Laufzeit in *expliziter* Form (als Polynom) und in der *O-Notation* an.

Die allgemeine Summenformel:

$$\sum a_i = a_1 m + \frac{m(m-1)}{2} d$$

wobei: m = Anzahl Summanden in der Summe
 d = Abstand zwischen den Summanden

Aufgabe 6 (O-Notation)

Bestimmen Sie das Laufzeitverhalten (Anzahl Multiplikationen und Additionen) für die Berechnung des Produktes zweier $n \times n$ Matrizen. Begründen Sie Ihre Aussage.

Allgemeine Berechnung:

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \cdot \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{pmatrix} = \begin{pmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{pmatrix} = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}$$

Beispiel:

$$\begin{pmatrix} 3 & 4 \\ 2 & 5 \end{pmatrix} \cdot \begin{pmatrix} 6 & 1 \\ 7 & 8 \end{pmatrix} = \begin{pmatrix} 3 \cdot 6 + 4 \cdot 7 & 3 \cdot 1 + 4 \cdot 8 \\ 2 \cdot 6 + 5 \cdot 7 & 2 \cdot 1 + 5 \cdot 8 \end{pmatrix} = \begin{pmatrix} 46 & 35 \\ 47 & 42 \end{pmatrix}$$

Geben Sie die Laufzeit in *expliziter* Form (als Polynom) und in der *O-Notation* an.

Aufgabe 7 (Konzepte)

Es wird eine generelle Rundungs-Funktionalität benötigt, mit welcher man auf beliebige Werte runden kann, z.B. auf eine Halbe- oder Viertel-Note, 5 Rappen oder 15 Minuten.

Nachfolgendes Szenario soll die Benutzung definieren:

```
System.out.println("Runden auf Halbe-Note:");
System.out.println("  5.24 -> " + round(5.24, 0.5));
System.out.println("  5.25 -> " + round(5.25, 0.5));
System.out.println("Runden auf Viertel-Note:");
System.out.println("  5.12 -> " + round(5.12, 0.25));
System.out.println("  5.13 -> " + round(5.13, 0.25));
System.out.println("Runden auf 5 Rappen:");
System.out.println("  72 -> " + round(72, 5));
System.out.println("  73 -> " + round(73, 5));
System.out.println("Runden auf 15 Minuten:");
System.out.println("  22 -> " + round(22, 15));
System.out.println("  23 -> " + round(23, 15));
```

Der erwartete Session-Log auf der Konsole:

```
Runden auf Halbe-Note:
  5.24 -> 5.0
  5.25 -> 5.5
Runden auf Viertel-Note:
  5.12 -> 5.0
  5.13 -> 5.25
Runden auf 5 Rappen:
  72 -> 70
  73 -> 75
Runden auf 15 Minuten:
  22 -> 15
  23 -> 30
```

Erstellen Sie ein entsprechendes Programm.