Algorithmen und Datenstrukturen
Aufgabenblatt 01

Abgabe bis: 29.04.2021 22:00 Uhr

Bitte bearbeiten Sie die Aufgaben in Dreiergruppen.

1. Laufzeitverhalten

Wie oft wird bei den folgenden Programmfragmenten die if-Abfrage ausgeführt, wie oft das print? Entwickeln Sie Formeln dafür.

```
i = 0
while i <= n
    if i mod 3 = 0
        print(i)
    i++
```

Die if-Abfrage wird $\sum_{i=0}^{n} 1 = n+1$ ausgeführt.
Das print wird $\sum_{i=0}^{n} \frac{1}{3} = \frac{n+1}{3}$ ausgeführt.

```
i = j
while i <= n
    if i mod 3 = 0
        print(i)
    i++
```

Die if-Abfrage wird $\sum_{i=j}^{n} 1 = n-j+1$ ausgeführt.
Das print wird $\sum_{i=j}^{n} \frac{1}{3} = \frac{n-j+1}{3}$ ausgeführt.

```
for i = a to b (b ausschließlich)
    for j = 2 to 4 (4 ausschließlich)
        if j * i gerade
            print(j, i)
```

Die if-Abfrage wird $\sum_{i=a}^{b-1} \sum_{j=2}^{3} 1 = \sum_{i=a}^{b-1} 2 = (b-a) \cdot 2$ ausgeführt.

Das print wird $\sum_{i=a}^{b-1} \sum_{j=2}^{3} \frac{3}{4} = \sum_{i=a}^{b-1} \frac{6}{4} = \frac{3}{2} \cdot (b-a)$

Ziel: Vorberechnung zur Laufzeitermittlung

2. Laufzeitverhalten

Welche Befehle werden in dem folgenden Pseudocodefragment am häufigsten ausgeführt und wie oft? Entwickeln Sie eine Formel in Abhängigkeit von n.

Die Befehle print(i, j i*j); print(newline); if j ist gerade Zahl werden am häufigsten ausgeführt.

```
input(n)
for i = 1 to n
    for j = i + 1 to n
        print(i, j i*j)
        print(newline)
        if j ist gerade Zahl
            print("gerade")
            print(newline)
```

Formel: $\sum_{i=1}^{n} \sum_{j=1+1}^{n} 1 = \sum_{i=1}^{n} n-i+1$

Ziel: Vorberechnung zur Laufzeitermittlung

3. Laufzeitverhalten, Implementierung

Implementieren Sie den intuitiven Pseudocode-Algorithmus *maxTeilsumme3* aus der Vorlesung (Folie 19) in Python. Benutzen Sie dabei keine objektorientierte Programmierung. Zählen Sie in einer zusätzlichen Variablen die Anzahl der Additionen in der Zeile

```
summe += a[k]
```

Ermitteln Sie die Laufzeit der Hauptschleife des Programms in Sekunden, z. B. mit *time.time()*.

Eingabe: eine Folge A von Integer-Zahlen in einer vom Benutzer auszuwählenden Textdatei (zur Laufzeit)
Ausgaben: die maximale Teilsumme von A
die Indices des ersten und letzten Elementes der Teilfolge von A mit der maximalen Teilsumme
die Anzahl der Additionen
die benötigte Zeit

Ziel: Umsetzung eines Algorithmus in Python gemäß Spezifikation

4. Laufzeitverhalten, Implementierung

Implementieren Sie denselben Algorithmus in Java (möglicherweise mit OOP).

Ziel: Umsetzung eines Algorithmus in Java gemäß Spezifikation

5. Auswertung

Bereiten Sie drei alternative Eingabedateien vor mit unterschiedlich langen Eingabefolgen A mit 30, 300 und 3000 Werten.
Führen Sie jedes der beiden Programme mit jeder Ihrer Eingabedateien aus und tragen die Ergebnisse in folgenden Tabelle ein, bzw. berechnen die gewünschten Werte (siehe Vorlesung):

n	n^3	$\frac{1}{6}n^3+\frac{1}{2}n^2+\frac{1}{3}n$	Additionen (Python)	Laufzeit (Python)	Additionen (Java)	Laufzeit (Java)
30	27000	4960	4960	0,001 Sekunden	4960	0,001 Sekunden
300	27000000	4545100	4545100	0,2 Sekunden	4545100	0,016 Sekunden
3000	27000000000	4504501000	4504501000	256,3 Sekunden	4504501000	4,17 Sekunden

Interpretieren Sie Ihre Ergebnisse.
Haben Sie diese Werte erwartet?
Ziel: Interpretation von Daten über das Laufzeitverhalten eines Programms.

6. Spezifikationen

Schreiben Sie je eine Spezifikation für einen Algorithmus und für ein Programm zum Finden der kleinsten Zahl in einer Folge.
Ziel: Schreiben einer Spezifikation

5)

Aus den Ergebnissen ist zu interpretieren, dass Java ein schnelleres Laufzeitverhalten als Python hat und die Anzahl der zu benötigten Sekunden exponentiell im Verhältnis zu n steigt.
Wir haben erwartet, dass Java schneller als Python ist, da Java eine Interpretierte Sprache und Python eine Kompilierte Sprache ist. Python wird erst zur Laufzeit kompiliert und braucht dadurch länger für die gleiche Anzahl an Elementen.

6)

Spezifikation für Algorithmus:
Eingabe: Folge von Zahlen in einer Liste
Ausgabe: kleinste Zahl aus der Folge

Spezifikation für Programm:

- Das Programm soll aus einer Folge von Zahlen, die kleinste Zahl ausgeben. Die Zahlenfolge soll aus einer Textdatei eingelesen werden, wobei jede Zeile eine Zahl beinhalten soll. Die kleinste Zahl soll in der Konsole ausgegeben werden.
- Programmname: findSmallestNumber