

Algorithmen und Datenstrukturen

Aufgabenblatt 01

Abgabe bis: 19.04.2023 17:45 Uhr
Punkte: 20

1. Laufzeitverhalten

2 Pkt.

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

```
i = j
while i <= n // gehen Sie davon aus, dass i=j<n ist
    if i mod 3 = 0
        print(i)
    i++

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

Ziel: Vorberechnung zur Laufzeitermittlung

2. Laufzeitverhalten

2 Pkt.

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

```
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)
```

Ziel: Vorberechnung zur Laufzeitermittlung

3. Laufzeitverhalten, Implementierung

6 Pkt.

Implementieren Sie den intuitiven Pseudocode-Algorithmus *maxTeilsomme3* 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

6 Pkt.

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

Ziel: Umsetzung eines Algorithmus in Java gemäß Spezifikation

5. Auswertung

2 Pkt.

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	4495	3,9ms	4495	1ms
300	27000000	4545100	4499950	625ms	4499950	32ms
3000	27000000000	4504501000	205032204	10+min	205032204	4552ms

Interpretieren Sie Ihre Ergebnisse.

Haben Sie diese Werte erwartet?

Ziel: Interpretation von Daten über das Laufzeitverhalten eines Programms.

6. Spezifikationen

2 Pkt.

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

Ziel: Schreiben einer Spezifikation