

Universität Potsdam
Institut für Informatik
Algorithmen und Datenstrukturen

05. Aufgabenblatt

Lernziele (zum Abhaken): Die Student:innen können...

- die Zeit- sowie Platzkomplexität eines gegebenen iterativen Algorithmus bestimmen.
- die Zeitkomplexität eines gegebenen rekursiven Algorithmus mithilfe der Rekursionsgleichung bestimmen.
- die Platzkomplexität eines gegebenen rekursiven Algorithmus bestimmen.
- die Komplexität eines, durch eine Rekursionsgleichung gegebenen, rekursiven Algorithmus mithilfe des Master-Theorems bestimmen.
- einen gegebenen Teile-und-Herrsche-Algorithmus unter Verwaltung der Parameter und Visualisierung der Zwischenergebnisse anwenden.

8 Komplexität rekursiver Algorithmen

1. Schreiben Sie ein Python-Programm, dass die Funktion

$$f : \mathbb{N} \rightarrow \mathbb{N} \text{ mit } f(n) = n!$$

- (a) iterativ berechnet,
 - (b) rekursiv berechnet.
2. Analysieren Sie die Laufzeit der beiden Programme. Für das rekursive Programm stellen Sie auch die Rekursionsgleichung auf und lösen diese schrittweise (ohne Master-Theorem).
3. Analysieren Sie den Bedarf an Platz im Hauptspeicher für beide Programme.
4. Geben Sie die Komplexität der Algorithmen an (in O -Notation), wenn folgende Rekursionen dafür gefunden werden können.

(a) $t(n) = 3t(\frac{n}{2}) + n$

(b) $t(n) = 3t(\frac{n}{2}) + n^2$

(c) $t(n) = 8t(\frac{n}{2}) + n^3$

Hinweis: Bei der Teilaufgabe 4 dürfen Sie das Master-Theorem anwenden.

9 Maximale Teilsumme

Berechnen Sie die maximale Teilsumme der Sequenz

$$(4, -3, -2, 5, 2, 1, -5, 7)$$

1. nach dem Brute Force Algorithmus (`maxTeilsumme_1`),
2. nach dem Teile-und-Herrsche-Algorithmus.
Notieren Sie alle Aufrufe mit allen Parametern und stellen Sie alle Zwischenergebnisse dar.