

Algorithmen und Datenstrukturen

Aufgabe 1 **Laufzeiten**

Gegeben seien Algorithmen A_1 , A_2 und A_3 , welche die Laufzeiten

$$T_1(n) = c_1 n, \quad T_2(n) = c_2 n^3 \quad \text{und} \quad T_3(n) = c_3 2^n$$

haben, wobei die $c_i, i = 1, 2, 3$, konstant sind. Für jeden Algorithmus bezeichne $m_i, i = 1, 2, 3$, die maximale Größe der Eingabe, die innerhalb einer fest vorgegebenen Zeit T auf einem Rechner R verarbeitet werden kann.

Wie ändern sich die Zahlen $m_i, i = 1, 2, 3$, wenn der Rechner R durch einen k -mal schnelleren Rechner ersetzt wird?

Aufgabe 2 **Maximale Teilsequenz - Bruteforce**

- a) **Implementierung.** Schreiben Sie eine C/C++ Funktion, die als Eingabe einen `vector<int>` A bekommt und den Wert der maximalen Teilsequenz $y(A)$ berechnet.

$$y(A) = \max_{i \leq j} \sum_{k=i}^j x_k$$

Ihr Algorithmus muss nicht die optimale Laufzeit haben. Nutzen Sie gerne auch das Testing Framework¹.

- b) **Analyse.** Geben Sie die Anzahl der Grundoperationen an, die Ihr Algorithmus abhängig von der Länge der Eingabe n benötigt (analog der Schulmethode für die Multiplikation von Zahlen).

Als Grundoperationen betrachten wir Multiplikation und Addition von Integern.

Aufgabe 3 (P) **Exam Room Order**

Ein Prüfungsraum besteht aus einer Reihe von n Sitzplätzen mit den Aufschriften von 0 bis $n - 1$.

Jedem neu ankommenden Prüfling wird ein fixer Platz zugewiesen. Dieser Platz ist immer so zu wählen, dass der Abstand zu dem nächstgelegenen besetzten Platz maximal ist. Sollte es mehrere solche Plätze geben, wird der mit der niedrigsten Nummer gewählt. Der erste zu besetzende Platz ist die Nummer 0.

In dieser Aufgabe implementieren Sie eine Funktion, die diese Reihenfolge für ein gegebenes n berechnet. Nutzen Sie gerne auch das Testing Framework¹.

¹ Weitere Informationen in *Setup Programmieraufgaben* in Moodle

- a) Überlegen Sie eine geeignete Datenstruktur, um die besetzten Plätze zu speichern. Beachten Sie dabei insbesondere diese Fragen:
- Was ist über Wertebereich/maximale Anzahl der Datenpunkte bekannt?
 - Welche Operationen soll die Datenstruktur zum Einfügen/Auslesen anbieten? Welche Informationen über die Daten sind wichtig?
 - Hilft es, die Daten sortiert abzuspeichern?
 - Gibt es bereits eine Implementierung in der C++ Standard Library?
- b) Schreiben Sie eine Funktion `nextSeat`, die aus Ihrer gewählten Datenstruktur die Nummer des nächsten zu besetzenden Platzes berechnet.
- c) Implementieren Sie eine Funktion `examRoomOrder`, welche die gesamte Reihenfolge der vergebenen Sitzplätze für ein gegebenes n berechnet. Benutzen Sie dafür ihre gewählte Datenstruktur und die Funktion `nextSeat`.