

Algorithmen und Datenstrukturen

Aufgabe 1 Asymptotische Notation I

Vergegenwärtigen Sie sich zunächst die Definitionen von $O(n)$ und $\Theta(n)$, wie sie in der Vorlesung vorgestellt wurden. Zeigen Sie dann folgende Behauptungen mit einem formalen Beweis wie in der Vorlesung (Grenzwertabschätzung oder Mengendarstellung):

- a) $7n^4 \in O(n^5)$
- b) $n^2/2 - 2n \in \Theta(n^2)$
- c) $\log n \in o(n)$
- d) $2^{2n} \notin O(2^n)$

Aufgabe 2 Asymptotische Notation II

Treffen die folgenden Behauptungen zu? Beweisen sie deren Richtigkeit, oder widerlegen Sie sie, entweder mittels Beweis oder durch Angabe eines Gegenbeispiels!

- a) Sei die Komplexität einer Funktion f bestimmt als $O(n)$. Dann ist die n -malige Ausführung $O(n^2)$.
- b) Falls $f(n) = \Theta(g(n))$, dann folgt $2^{f(n)} \in \Theta(2^{g(n)})$.
- c) $n^n = O(2^n)$
- d) Falls $g = O(f)$ und $h = O(f)$, dann gilt auch $g = O(h)$

Aufgabe 3 (P) Addition

Gegeben sind eine Basis b und zwei `vector<int>` x , y , die zwei positive base- b Integer repräsentieren. Ziffer i ist dabei an Stelle i mit Wertigkeit b^i , geordnet von der höchst- zur niedrigstwertigen Ziffer von Anfang bis Ende der Liste, die höchstwertige Ziffer steht also am Anfang der Liste. Es gibt keine führenden Nullen.

Bsp.: $[42, 53, 28], [37], 55 \rightarrow [42, 54, 10]$

- a) Implementieren Sie eine C++ Funktion, um die repräsentierten Zahlen zu addieren und die Summe als `vector<int>` zurückzugeben. Beachten Sie, dass die Listen der Zahlen unterschiedlich lang sein können. Nutzen Sie gerne auch das Testing Framework¹.

¹ Weitere Informationen in *Setup Programmieraufgaben* in Moodle

Als optionale Hilfestellung finden Sie in der Datei `blatt2hint.cpp` ein grobes Gerüst einer Lösung, welches Sie verwenden können, wenn Sie nicht weiter kommen.

Hinweis: Vielleicht hilft es Ihnen, die Zahlen zu Beginn so zu tauschen, dass z.B. y die Zahl mit mehr Ziffern ist.

- b) Seien n_x, n_y jeweils die Anzahl der Ziffern der Zahlen x, y geschrieben zur Basis b . Geben Sie abhängig von n_x, n_y eine Abschätzung an die worst-case Laufzeit Ihrer Implementierung in asymptotischer Notation an.