

Algorithmen und Datenstrukturen (WS 2019)

Aufgabenblatt 1

zu bearbeiten bis: 11.11.2019 / 13.11.2019

Aufgabe 1.1 (Rekursion - Theorie)

Wir definieren die folgende rekursive Funktion $f_{oo}(a, b)$ (a und b seien ganze Zahlen):

$$\begin{aligned}f_{oo}(0, b) &= 0 \\f_{oo}(a, b) &= bar(b, a, b) \\bar{a}(0, a, b) &= f_{oo}(a - 1, b) \\bar{a}(i, a, b) &= a + \bar{a}(i - 1, a, b)\end{aligned}$$

- Was ist der Definitionsbereich von Funktion $f_{oo}()$?
(Also für welche Eingabewerte $(a, b) \in \mathbb{Z}^2$ terminiert die Funktion?)
- Was berechnet die Funktion $f_{oo}()$? Vereinfachen Sie die Formel möglichst weit.

Aufgabe 1.2 (Primzahlen - Theorie)

Algorithmus 1 wird als “Sieb des Eratosthenes” bezeichnet. Dieser bestimmt alle Primzahlen unterhalb einer gegebenen Grenze $N \geq 2$:

Algorithmus 1 Sieb des Eratosthenes

- procedure** SIEB(N)
 - Schreibe alle Zahlen von 2 bis N auf.
 - Markiere die Zahl 2 als Primzahl und streiche alle Vielfachen der 2.
 - Finde die kleinste nicht gestrichene und nicht markierte Zahl m .
 - Markiere m als Primzahl und streiche alle Vielfachen von m .
 - Führe die Schritte 4-5 solange aus, bis keine Zahl $m \leq N$ mehr gefunden wird.
 - Alle markierten Zahlen sind Primzahlen.
-

- Führen Sie den Algorithmus für $N=17$ durch und zeichnen Sie den Verlauf tabellarisch auf.
- Terminiert der Algorithmus immer? Ist er determiniert? Ist er deterministisch? Ist er korrekt? Begründen Sie jeweils.
- Sind Berechnungen in Java immer deterministisch? Begründen Sie.
- Ab welcher Grenze $g < N$ können wir uns das Streichen der Vielfachen in Schritt 5 sparen? Begründen Sie.

Aufgabe 1.3 (Primzahlen - Praxis)

Implementieren Sie das Verfahren, sowie zwei Hilfsmethoden für die Ausgabe (Sie können zuerst b) und c) bearbeiten, um die Methoden dann beim Entwickeln von a) zum Ausprobieren verwenden zu können):

- a) Implementieren Sie das Sieb des Eratosthenes mit einer Methode `boolean[] sieve(int N)`. Diese soll ein Boolesches Array b der Länge $N + 1$ zurückliefern, das für jede Zahl $k = 0, \dots, N$ anzeigt ob es sich um eine Primzahl handelt (`b[k] == true`) oder nicht (`b[k] == false`).
- b) Implementieren Sie eine Methode `int count(boolean[] array)`, welche die Anzahl der `true`-Werte in einem Array zählt.
- c) Implementieren Sie eine Methode `int[] up_to(int n)`, welche die beiden Methoden `sieve` und `count` verwendet, um ein Array der Primzahlen bis einschließlich n zu erzeugen.
- d) Wie viele Primzahlen ≤ 10.000 gibt es?