Prof. Dr. Klaus Volbert

Regensburg, den 03. Oktober 2018

Votieren und abgeben bis zum 09. Oktober 2018, 23:00 Uhr

Übungen zur Vorlesung

Algorithmen und Datenstrukturen

WiSe 2018/19

Blatt 1

Wichtige Hinweise:

– > Falls Sie bei der Bearbeitung einer Aufgabe größere Schwierigkeiten hatten und deswegen die Bearbeitung abgebrochen haben, so versuchen Sie bitte Ihre Schwierigkeiten in Form von Fragen festzuhalten. Bringen Sie Ihre Fragen einfach zur Vorlesung oder zur Übung mit!

– > Kursraum: https://elearning.uni-regensburg.de/course/view.php?id=9228

Aufgabe 1:
Implementieren Sie den in der Vorlesung vorgestellten euklidischen Algorithmus zur Bestimmung des größten gemeinsamen Teilers (ggT) zweier natürlicher Zahlen in einer Programmiersprache Ihrer Wahl (C, C++, Java oder C#). Ergänzen Sie eine rekursive Implementierung des Algorithmus. Welchen Algorithmus halten Sie wann für sinnvoller? Implementieren Sie einen **eigenen** Algorithmus zur Bestimmung des kleinsten gemeinsamen Vielfachen (kgV) zweier natürlicher Zahlen. Geben Sie für $a, b \in \{30, \ldots, 40\}$ die Werte $\mathrm{kgV}(a, b)$, $\mathrm{ggT}(a, b)$ und $a \cdot b$ aus. Was fällt Ihnen auf?

Aufgabe 2:
In der Vorlesung wurde das Sieb des Eratosthenes vorgestellt, mit dem alle Primzahlen bis zu einem gegebenen Wert k (obere Schranke) bestimmt werden können.

1. Ermitteln Sie anhand einer dem Sieb des Eratostehenes entsprechenden Tabelle durch Einkreisen und Streichen alle Primzahlen bis zum Wert 100.

2. Implementieren Sie einen Algorithmus in einer Programmiersprache Ihrer Wahl (C, C++, Java oder C#), der die Tabelle unter Eingabe einer oberen Schranke k ausgibt und lassen Sie danach alle Primzahlen bis zum Wert k ausgeben.

3. Geben Sie mit Ihrem Algorithmus alle Primzahlen bis zum Wert 100000 aus.

Aufgabe 3:
Definieren Sie einen abstrakten Datentyp Matrix für ganzzahlige $m \times n$ Matrizen mit folgenden Operationen:

- `Init()`: Initialisiert die Matrix mit Wert 0
- `Print()`: Gibt die Matrix am Bildschirm aus

- `Input()`: Liest die Matrix von der Tastatur ein
- `Add(M)`: Liefert die Summe aus der Matrix und M
- `Mult(M)`: Liefert das Produkt aus der Matrix und M

Implementieren Sie den abstrakten Datentyp Matrix als Datenstruktur in einer Programmiersprache Ihrer Wahl (C, C++, Java oder C#) und demonstrieren Sie die Funktionsweise.

Erinnerung: Addition zweier Matrizen:

- Eingabe: $m, n \in \mathbb{N}$, $A = (a_{ij})_{i=1,\ldots,m,j=1,\ldots,n}$ und $B = (b_{ij})_{i=1,\ldots,m,j=1,\ldots,n}$ mit $a_{ij}, b_{ij} \in \mathbb{Z}$
- Ausgabe: $C = (c_{ij})_{i=1,\ldots,m,j=1,\ldots,n}$ mit $c_{ij} = a_{ij} + b_{ij}$

Erinnerung: Multiplikation zweier Matrizen:

- Eingabe: $l, m, n \in \mathbb{N}$, $A = (a_{ij})_{i=1,\ldots,l,j=1,\ldots,m}$ und $B = (b_{ij})_{i=1,\ldots,m,j=1,\ldots,n}$ mit $a_{ij}, b_{ij} \in \mathbb{Z}$
- Ausgabe: $C = (c_{ij})_{i=1,\ldots,l,j=1,\ldots,n}$ mit $c_{ij} = \sum_{k=1}^{m} a_{ik} \cdot b_{kj}$

Aufgabe 4:
Erweitern Sie Ihre Implementierung aus Aufgabe 3 dadurch, dass Sie für die beiden Operationen `Add` und `Mult` jeweils ausgeben, wie viele Additionen und Multiplikationen durchgeführt wurden (Anzahl) und wieviel Zeit für die Durchführung notwendig war (in ms). Ermitteln Sie durch Ergänzung von geschickten Test-Operationen, wie groß quadratische Matrizen werden dürfen, wenn für eine Addition bzw. eine Multiplikation 1, 2, 5 bzw. 10 Minuten Verarbeitungszeit zur Verfügung stehen.