Prof. Dr. Klaus Volbert Regensburg, den 23. Oktober 2019

Votieren und abgeben bis zum 29. Oktober 2019, 23:00 Uhr

Übungen zur Vorlesung

Algorithmen und Datenstrukturen

WiSe 2019/20

Blatt 4

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 die Funktion $f : \mathbb{N}_0 \times \mathbb{N}_0 \to \mathbb{N}$ mit und ohne Verwendung von Rekursion (Tipp: Verwenden Sie einen Stapel für natürliche Zahlen für die iterative Implementierung):

$$f(n,m) = \begin{cases} m+1, & \text{für } n = 0 \\ f(n-1,1), & \text{für } m = 0 \text{ und } n \geq 1 \\ f(n-1, f(n, m-1)), & \text{sonst} \end{cases}$$

Zeigen Sie mittels vollständiger Induktion, dass $f(n,m)$ für alle $n,m \in \mathbb{N}_0$ definiert ist. Ermitteln Sie anhand Ihrer Implementierung möglichst viele Werte $f(n,m)$.

Aufgabe 2:
Geben Sie jeweils an, welche Beziehung im O-Kalkül (O,Ω,Θ)zwischen den jeweiligen Funktionen gilt.

$$f(n) = \sum_{i=0}^{n}(2i+1) \quad , \quad g(n) = 3n^2 + 17 \tag{1}$$

$$f(n) = \log(n!) \quad , \quad g(n) = 3n\log n^{(5n)} \tag{2}$$

$$f(n) = 2^{n+5} + n^2 \quad , \quad g(n) = \sum_{i=0}^{n} 2^i \tag{3}$$

$$f(n) = 3n^2 \quad , \quad g(n) = 9^{\log_3(n)} \tag{4}$$

Aufgabe 3:
Stellen Sie sich vor, Sie sollen zwei quadratische Matrizen $M, N \in \mathbb{R}^{n\times n}$ miteinander multiplizieren. Sei $n = 2^i$ für ein $i \in \mathbb{N}$, dann kann man M, N und $O = M \cdot N$ wie folgt zerlegen mit $M_{ij}, N_{ij}, O_{ij} \in \mathbb{R}^{n/2\times n/2}$ für $i,j \in \{1,2\}$:

$$M := \begin{pmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{pmatrix}, N := \begin{pmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{pmatrix}, O := \begin{pmatrix} O_{11} & O_{12} \\ O_{21} & O_{22} \end{pmatrix}$$

Zeigen Sie, dass die beiden folgenden Varianten die Produktmatrix O korrekt berechnen:

- Variante 1:

$$
\begin{aligned}
O_{11} &:= M_{11} \cdot N_{11} + M_{12} \cdot N_{21} \\
O_{12} &:= M_{11} \cdot N_{12} + M_{12} \cdot N_{22} \\
O_{21} &:= M_{21} \cdot N_{11} + M_{22} \cdot N_{21} \\
O_{22} &:= M_{21} \cdot N_{12} + M_{22} \cdot N_{22}
\end{aligned}
$$

- Variante 2:

$$
\begin{aligned}
H_1 &:= (M_{11} + M_{22}) \cdot (N_{11} + N_{22}) \\
H_2 &:= (M_{21} + M_{22}) \cdot N_{11} \\
H_3 &:= M_{11} \cdot (N_{12} - N_{22}) \\
H_4 &:= M_{22} \cdot (N_{21} - N_{11}) \\
H_5 &:= (M_{11} + M_{12}) \cdot N_{22} \\
H_6 &:= (M_{21} - M_{11}) \cdot (N_{11} + N_{12}) \\
H_7 &:= (M_{12} - M_{22}) \cdot (N_{21} + N_{22})
\end{aligned}
$$

$$
\begin{aligned}
O_{11} &:= H_1 + H_4 - H_5 + H_7 \\
O_{12} &:= H_3 + H_5 \\
O_{21} &:= H_2 + H_4 \\
O_{22} &:= H_1 - H_2 + H_3 + H_6
\end{aligned}
$$

Bestimmen Sie die asymptotische Laufzeitkomplexität beider Varianten und vergleichen Sie diese mit der Komplexität der Standardmethode zur Multiplikation zweier Matrizen.

Aufgabe 4:
Implementieren Sie die Variante 2 aus Aufgabe 3 in C, C++, C# oder Java. Ist Ihre Implementierung schneller als die Standard-Methode? Wenn ja, ab welcher Dimension n?