Prof. Dr. Frank Noé
Dr. Christoph Wehmeyer
Tutoren:
Katharina Colditz; Anna Dittus;
Felix Mann; Christopher Pütz

7. Übung zur Vorlesung
Computerorientierte Mathematik I

Abgabe: Freitag, 12.12.2014, 16:00 Uhr, Tutorenfächer Arnimallee 3

`http://www.mi.fu-berlin.de/w/CompMolBio/ComaI`

Aufgabe 1 *(Komplexität, 2T)*:

Sei $\mathbf{A} \in \mathbb{R}^{n \times n}$ eine quadratische Matrix und $\mathbf{x} \in \mathbb{R}^n$ ein Vektor. Zeigen Sie, dass die Komplexität der Matrix-Vektor-Multiplikation $\mathbf{y} = \mathbf{A}\mathbf{x}$ höchstens quadratisch in der Dimension n ist. Dabei sei das Aufwandsmaß die Anzahl der nötigen Multiplikationen und Additionen.

Aufgabe 2 *(Sortieren, 8T)*:

Ein weiterer Sortieralgorithmus ist **Insertion Sort**. Dieses Verfahren lässt sich folgendermaßen beschreiben (Eingabe sei eine Liste $\mathbf{x}$ der Länge $n \geq 2$, wie gewohnt bezeichnen wir mit $\mathbf{x}(i)$ das i-te Element in der Liste):

1. Setze $i = 2$.
2. Setze $j = i$. Solange $j > 1$ und $\mathbf{x}(j-1) > \mathbf{x}(j)$ gelten, wiederhole:
   (a) Vertausche $\mathbf{x}(j-1)$ und $\mathbf{x}(j)$.
   (b) Setze $j = j - 1$.
3. Falls $i = n$ ist, breche ab. Sonst setze $i = i + 1$ und wiederhole Schritt 2.

a) *(3T)* Begründen Sie, dass Insertion Sort korrekt ist, also dass am Ende die Liste $\mathbf{x}$ aufsteigend sortiert ist.
b) *(3T)* Zeigen Sie, dass Insertion Sort quadratische Laufzeit besitzt, also dass

$$T_A(n) \quad = \quad \mathcal{O}(n^2).$$

Dabei misst die Eingabegröße n die Länge der Liste $\mathbf{x}$ und das Aufwandsmaß ist die Anzahl der benötigten Vergleiche.
c) *(2T)* Sei $\mathbf{x}$ eine Liste der Länge n, die bereits aufsteigend sortiert ist. Wie viele Vergleiche benötigt Insertion Sort, und wie viele benötigt Bubble Sort?

Aufgabe 3 *(Euklidischer Algorithmus, 5P)*:

a) *(2P)* Schreiben Sie eine Matlab-Funktion, welche für zwei natürliche Zahlen $a \geq b \geq 1$ den größten gemeinsamen Teiler $\mathrm{ggT}(a, b)$ mittels des Euklidischen Algorithmus bestimmt, und außerdem noch die Anzahl der Schritte im Euklidischen Algorithmus zählt und zurückgibt.
b) *(3P)* Wählen Sie nacheinander $b = 10, 20, 50, 100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000$ und erzeugen Sie für jeden Wert von b jeweils 500 zufällige natürliche Zahlen $b \leq a \leq 10b$. Bestimmen Sie für jedes b die mittlere Anzahl der Schritte, die der Euklidische Algorithmus zur Bestimmung von $\mathrm{ggT}(a, b)$ benötigt hat. Plotten Sie diese zusammen mit der oberen Schranke $\log_\Phi(b) + 1$ aus der Vorlesung.