

Aufgabe 12.1

In der Klasse `Det.java` wird die Determinante der Matrix

$$A = \begin{pmatrix} 0.1234564 & 0.6736254 & 0.7970818 \\ 0.8756363 & 0.7765389 & 1.6521752 \\ 0.8653839 & 0.8523753 & 1.7177592 \end{pmatrix}$$

nach der sarrusschen Regel berechnet.

- Was ist die Determinante der Matrix A ?
- Deckt sich die Ausgabe des Programms `Det` mit Ihrem Ergebnis aus (a)? Wo könnte ggfs. das Problem liegen?

(4 Punkte)

Aufgabe 12.2

Ein Graph $G = (V, E)$ ohne Kantengewichte sei als Adjazenzmatrix $A = A_G$ gegeben (A ist also eine Matrix mit Einträgen aus $\{0, 1\}$). Aus A bilden wir die Matrix \bar{A} , indem wir die 0-en auf der Hauptdiagonale alle durch eine 1 ersetzen. Im Folgenden sei für eine natürliche Zahl k mit \bar{A}^k die k -te Potenz der Matrix \bar{A} bezeichnet (also $\bar{A}^1 = \bar{A}$, $\bar{A}^2 = \bar{A} \cdot \bar{A}$, $\bar{A}^3 = \bar{A}^2 \cdot \bar{A}$ usw. wobei \cdot die übliche Matrixmultiplikation bezeichnet). Versuchen Sie zu argumentieren, dass ein Graph G genau dann zusammenhängend ist, wenn es ein $k_0 \in \mathbb{N} \setminus \{0\}$ gibt, für das alle Einträge in $\bar{A}_G^{k_0}$ positiv sind. Wie würden Sie \bar{A}_G^k für gegebenes k berechnen? Was ist die Laufzeit Ihres Algorithmus im worst case in Abhängigkeit von der Anzahl n der Knoten des Graphen G und dem Exponenten k ?

(6 Punkte)

Aufgabe 12.3

Geben Sie eine Klassendefinition `GraphNode` an, mit der man einen zusammenhängenden Graph beliebiger Größe verwalten kann. In der Klasse sollen die Zeiger auf weitere Knoten des Graphs sowie zugehörigen Kosten, die mit den Kanten des Graphs assoziiert sind, gespeichert werden.

(4 Punkte)

Zusatzaufgabe 12.4

Im Material zu diesem Aufgabenblatt ist die Klasse `JumpingJack.java` enthalten. Darin finden Sie die Methode:

```
public static int jump(int[] a)
```

Analysieren Sie die Laufzeit im worst case dieser Methode in Abhängigkeit von der Länge n des Arrays a . Geben Sie Ihr Ergebnis unter Verwendung der O -Notation an.

(4+Punkte)
