

Die Adjazenzmatrix

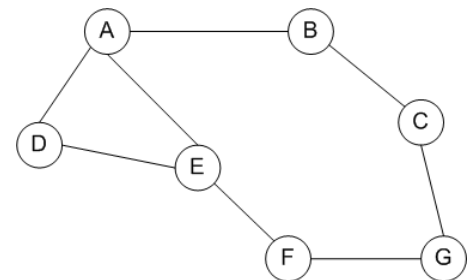
Eine **Adjazenzmatrix** ist eine Tabelle, deren Zeilen und Spalten jeweils durch die Knoten und deren Reihenfolge festgelegt sind. In den Zellen der Tabelle werden die Informationen zu den Kanten gespeichert:

Verbindet eine Kante zwei Knoten, so wird in der zugehörigen Zeile bei ungewichteten Graphen eine 1 eingetragen, und bei gewichteten Graphen das Kantengewicht.

Eine Adjazenzmatrix kann durch ein zweidimensionales Feld implementiert werden.

1. Ungewichteter ungerichteter Graph

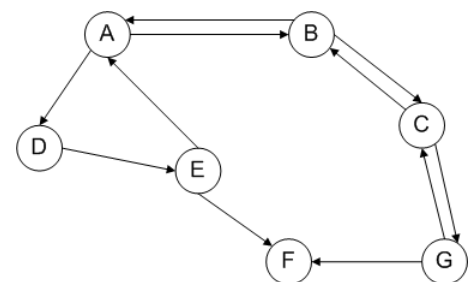
	A	B	C	D	E	F	G
A	0	1	0	1	1	0	0
B	1	0	1	0	0	0	0
C	0	1	0	0	0	0	1
D	1	0	0	0	1	0	0
E	1	0	0	1	0	1	0
F	0	0	0	0	1	0	1
G	0	0	1	0	0	1	0



Die Adjazenzmatrix ist symmetrisch zur Diagonalen.

2. Ungewichteter gerichteter Graph

	A	B	C	D	E	F	G
A	0	1	0	1	0	0	0
B	1	0	1	0	0	0	0
C	0	1	0	0	0	0	1
D	0	0	0	0	1	0	0
E	1	0	0	0	0	1	0
F	0	0	0	0	0	0	0
G	0	0	1	0	0	1	0

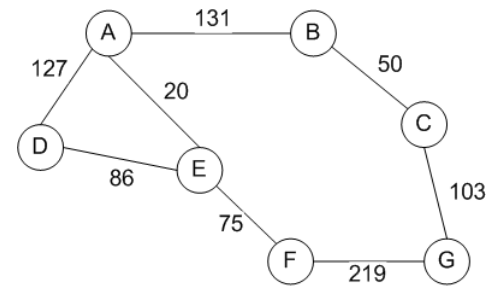


Die Adjazenzmatrix ist nicht symmetrisch zur Diagonalen.

Ist der Graph ungewichtet, so trägt man für existierende Kanten eine 1 ein und für nicht existierende eine 0.

3. Gewichteter ungerichteter Graph

	A	B	C	D	E	F	G
A	0	131	∞	127	20	∞	∞
B	131	0	50	∞	∞	∞	∞
C	∞	50	0	∞	∞	∞	103
D	127	∞	∞	0	86	∞	∞
E	20	∞	∞	86	0	75	∞
F	∞	∞	∞	∞	75	0	219
G	∞	∞	103	∞	∞	219	0



Die Adjazenzmatrix ist symmetrisch zur Diagonalen.

Für nicht existierende Kanten sind bei gewichteten Graphen mehrere Arten von Einträgen üblich:

- ∞
- größter möglicher Wert (in Java: Integer.MAX_VALUE)
- -1 (in unserer Implementierung)

Zweidimensionales Feld:

```
int[][] matrix;  
matrix = new int[7][7];
```

```
//Definition der Feldvariablen  
//Festlegen der Feldgröße
```

```
matrix[Zeilennummer][Spaltennummer] = Wert;  
Wertzuweisung (z. B. für Fall 3.):
```

```
//Wertzuweisung  
matrix[0][0] = 0;  
matrix[0][1] = 131;  
matrix[1][2] = 50;
```

```
matrix[Zeilennummer][Spaltennummer];
```

```
//Zugriff auf bestimmten Wert
```

Falls man ein Feld mit unterschiedlicher Zeilen- und Spaltenanzahl hat: z. B.

```
int[][] feld = new int[8][5];
```

```
feld.length;  
feld[0].length;
```

```
//Länge der 1. Dimension (also 8)  
//Länge der 2. Dimension (also 5)
```