Elemente der Graphentheorie

Definition 1:

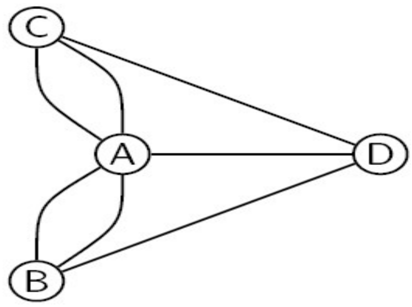
Ein **Graph** besteht aus **Knoten** (engl. vertex, vertices) und **Kanten** (engl. edge, edges). Die Menge der Knoten bezeichnen wir mit **V**, diejenige der Kanten mit **E**.

Der **Graph** heißt **zusammenhängend**, wenn er keine isolierte Knoten hat, d. h. wenn es zwischen je zwei Knoten stets eine Kante gibt.

Unter dem **Grad eines Knotens** verstehen wir bei einem ungerichteten Graph die Anzahl der Kanten, die von diesem Knoten ausgehen. (Hinweis: Bei gerichteten Graphen unterscheidet man zwischen Eingangsgrad und Ausgangsgrad.)

Zwei Knoten eines ungerichteten Graphen heißen **adjazent**, wenn sie durch eine Kante verbunden sind.

Beispiel für einen ungerichteten Graphen mit Mehrfachkanten:



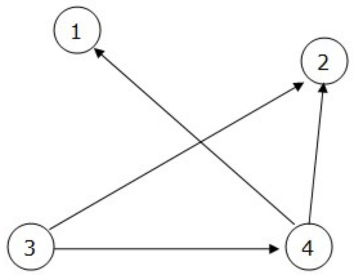
V = {A, B, C, D}

Grad(A) = 5
Grad(B) = Grad(C) = Grad(D) = 3

Definition 2:

Ein **Graph**, dessen Kanten in nur einer Richtung durchlaufen werden können, heißt gerichteter Graph.

Beispiel (Die Kanten werden als Pfeile dargestellt.):



V = {1, 2, 3, 4}

Eine von Knoten 4 nach Knoten 2 gerichtete Kante können wir als geordnetes Paar (4,2) beschreiben.

E = { (3,4), (4,2), (3,2), (4,1) }

Definition 3:

a) Unter einem **Eulerweg** verstehen wir einen Weg, der alle Kanten des ungerichteten Graphen genau einmal durchläuft.
b) Unter einem **Eulerkreis** verstehen wir einen geschlossenen Eulerweg, also einen Eulerweg, bei dem Start- und Zielknoten identisch sind.
c) Ein **Eulerscher Graph** ist ein Graph, der einen Eulerkreis besitzt.

Definition 4:

a) Unter einem **Hamiltonweg** verstehen wir einen Weg, der jeden Knoten des Graphen genau einmal besucht. Dabei dürfen Kanten auch mehrfach durchlaufen werden.
b) Unter einem **Hamiltonkreis** verstehen wir einen geschlossenen Hamiltonweg, also einen Hamiltonweg, bei dem Start- und Zielknoten identisch sind.
c) Ein **Hamiltonscher Graph** ist ein Graph, der einen Hamiltonkreis besitzt.

Satz von Euler (Leonard Euler 1736; strenger Beweis von Carl Hierholzer 1873):

Ein ungerichteter zusammenhängender Graph G hat einen Eulerweg genau dann, wenn die Anzahl der Knoten mit ungeradem Grad den Wert 0 oder 2 hat.

Ein ungerichteter zusammenhängender Graph G hat einen Eulerkreis genau dann, wenn jeder Knoten einen geraden Grad hat.

Die Struktur eines Graphen läßt sich mittels einer **Adjazenzmatrix** abbilden, welche die Verarbeitung durch einen Algorithmus ermöglicht.

Beispiel einer 3 x 3 – Matrix:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$a_{ij} \in R$
i = Zeilenindex, j = Spaltenindex

Die Komponente a_{23} steht in der 2. Zeile und der 3. Spalte.

Sind die Knoten des Graphen **G** mit **1**, **2**, **3**, ... , **n** bezeichnet, definieren wir:
Bei ungerichteten Graphen:

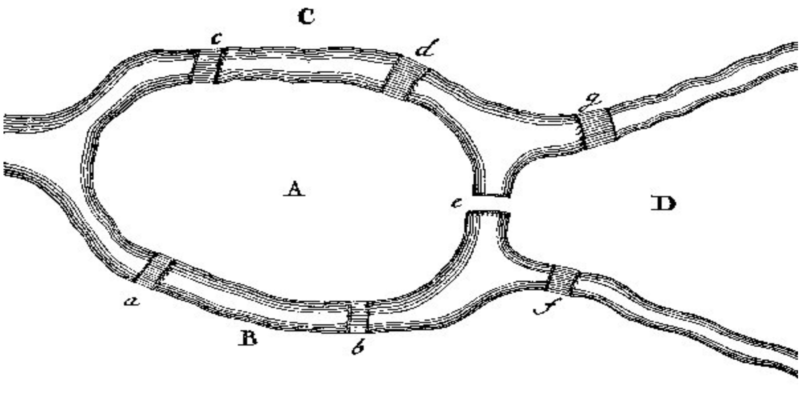
$\mathbf{a_{ij}}$ = Anzahl der Kanten zwischen den Knoten i und j
Falls der Graph Mehrfachkanten nicht enthält und die Kanten gewichtet sind (z. B. mit den Entfernungen der jeweiligen adjazenten Knoten i und j):
$\mathbf{a_{ij}}$ = Entfernung der adjazenten Knoten i und j

Bei gerichteten Graphen (ohne gleichgerichtete Mehrfachkanten):

$\mathbf{a_{ij}}$ = 1, falls die Kante von Knoten i in Richtung Knoten j verläuft;
$\mathbf{a_{ij}}$ = 0, falls die Kante von Knoten j in Richtung Knoten i verläuft oder keine Kante zwischen den Knoten i und j existiert.

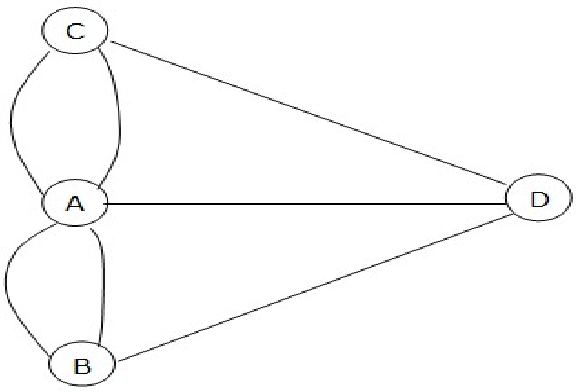
Beispiel 1

Das Königsberger Brückenproblem (1735)
Gibt es einen Weg, der jede der 7 Brücken a, b, . . . , g genau ein Mal überschreitet?



Äquivalente Formulierung:
Hat der Graph **G** mit der Knotenmenge **V** = {A, B, C, D} einen Eulerweg?

G



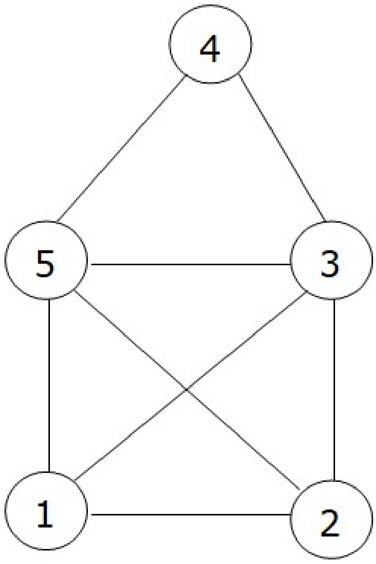
Wir ergänzen die **Adjazenzmatrix** um eine weitere, mit **#** bezeichnete Spalte, welche den Grad des jeweiligen Knotens angibt:

	A	B	C	D	#
A	0	2	2	1	5
B	2	0	0	1	3
C	2	0	0	1	3
D	1	1	1	0	3

Die Anzahl der Knoten mit ungeradem Grad beträgt 4 und hat daher weder den Wert 0 noch den Wert 2, folglich hat der Graph **G** keinen Eulerweg und erst recht keinen Eulerkreis.

Beispiel 2

Haus des Nikolaus



Adjazenzmatrix:

	1	2	3	4	5	#
1	0	1	1	0	1	3
2	1	0	1	0	1	3
3	1	1	0	1	1	4
4	0	0	1	0	1	2
5	1	1	1	1	0	4

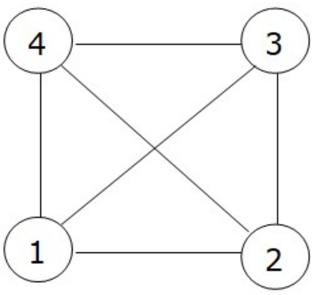
Da die Anzahl der Knoten mit ungeradem Grad den Wert 2 hat, gibt es nach dem Satz von Euler einen Eulerweg, aber keinen Eulerkreis.

Möglicher Eulerweg:

1 → 2 → 3 → 5 → 1 → 3 → 4 → 5 → 2

Beispiel 3

Quadrat mit Diagonalen



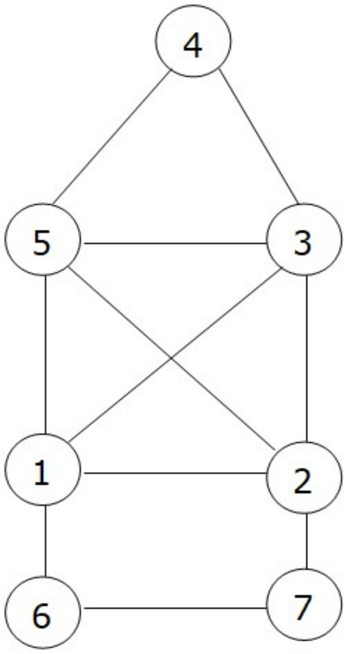
Adjazenzmatrix:

	1	2	3	4	#
1	0	1	1	1	3
2	1	0	1	1	3
3	1	1	0	1	3
4	1	1	1	0	3

Die Anzahl der Knoten mit ungeradem Grad beträgt 4 und hat daher weder den Wert 0 noch den Wert 2, folglich hat der Graph **G** keinen Eulerweg und erst recht keinen Eulerkreis.

Beispiel 4

Haus des Nikolaus mit Keller



Adjazenzmatrix:

	1	2	3	4	5	6	7	#
1	0	1	1	0	1	1	0	4
2	1	0	1	0	1	0	1	4
3	1	1	0	1	1	0	0	4
4	0	0	1	0	1	0	0	2
5	1	1	1	1	0	0	0	4
6	1	0	0	0	0	0	1	2
7	0	1	0	0	0	1	0	2

Da jeder Knoten einen geradzahligen Grad hat, gibt es nach dem Satz von Euler einen Eulerkreis.

Möglicher Eulerkreis:

$\mathbf{6 \rightarrow 7 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 1 \rightarrow 2 \rightarrow 5 \rightarrow 4 \rightarrow 3 \rightarrow 1 \rightarrow 6}$

Algorithmen zum Auffinden von Eulerwegen und –kreisen:

Wenn nach dem Satz von Euler die Existenz eines Eulerwegs oder einer Eulerkreises gesichert ist, liefern folgende Algorithmen nach Eingabe des Graphen (z. B. als Adjazenzmatrix) einen Eulerweg oder einen Eulerkreis:

Algorithmus von **Fleury**
Der Aufwand wächst mit dem Quadrat der Anzahl der Kanten, demnach hat der Algorithmus von Fleury quadratische Komplexität.

Algorithmus von **Hierholzer** (1873)
Der Aufwand wächst linear mit der Anzahl der Kanten, demnach hat der Algorithmus von Hierholzer lineare Komplexität.

Algorithmus zum Satz von Euler

Der Satz von Euler ist ein Existenzsatz, denn er stellt notwendige und hinreichende Kriterien bereit, um zu entscheiden, ob ein zusammenhängender ungerichteter Graph G einen Eulerweg oder einen Eulerkreis hat.

Als Beispiel nachfolgend ein in Pascal codierter Algorithmus, der nach Eingabe des Graphen G in Form der Adjazenzmatrix entscheidet, ob Graph G einen Eulerweg oder einen Eulerkreis oder keinen von beiden hat:

```
{
Nach Eingabe des Graphen G als Adjazenzmatrix entscheidet dieser Algorithmus,
ob G einen Eulerweg oder einen Eulerkreis oder keinen von beiden hat.
}

program euler;
uses crt;

type matrix = array[1..20,1..20] of integer;
     vektor = array[1..20] of integer;

var i, j, n, anzahl : integer;
    a : matrix;
    g : vektor;

begin
   clrscr;
   write('Anzahl (maximal 20) der Knoten: ');
   readln(n);

   {Eingabe der Adjazenzmatrix}
   for i:=1 to n do begin
                      for j:=1 to n do begin
                                        write('a(',i,',',j,') = ');
                                        readln(a[i,j])
                                       end
                      end;

   {Berechnung des Grades von jedem der n Knoten}
   for i := 1 to n do g[i] := 0;   {Initialisierung der Werte von g}
   for i := 1 to n do for j:=1 to n do g[i] := g[i] + a[i,j];

   {Ausgabe der Adjazenzmatrix}
   writeln;
   writeln('Adjazenzmatrix:');
   for i:=1 to n do begin
                      writeln;
                      for j:=1 to n do write(a[i,j],' ')
                    end;
   writeln;
   writeln;

   {Ausgabe des Grades eines jeden Knotens}
   writeln('Knotengrade:');
   for i:=1 to n do writeln(g[i]);
   writeln;

   {Entscheidung gemaess dem Satz von Euler}
   i := 0;
   anzahl := 0;
   repeat
     i := i+1;
     if odd(g[i]) then anzahl := anzahl + 1
   until i=n;

   if (anzahl=0) or (anzahl=2) then
     begin
       if anzahl=0 then begin
                          writeln('Der Grad jedes Knotens ist geradzahlig,');
                          writeln('folglich hat der Graph einen Eulerkreis.')
                        end;
       if anzahl=2 then begin
                          writeln('Der Grad von genau zwei Knoten ist ungerade,');
                          writeln('folglich besitzt der Graph einen Eulerweg,');
                          writeln('aber keinen Eulerkreis.')
                        end
     end
   else begin
          if anzahl=1 then writeln('Da ',anzahl,' Knoten einen ungeraden Grad hat,')
          else writeln('Da ',anzahl,' Knoten einen ungeraden Grad haben,');
          writeln('besitzt der Graph weder einen Eulerkreis noch einen Eulerweg.')
        end;

   while not keypressed do

end.
```