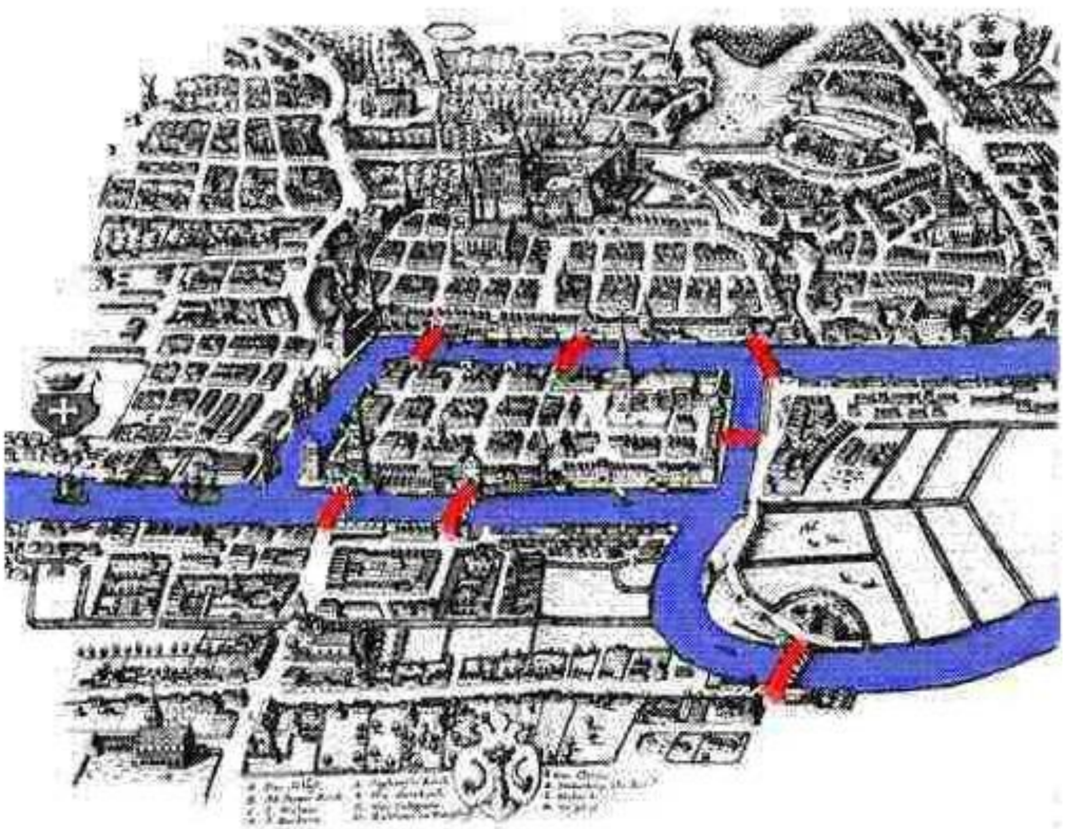
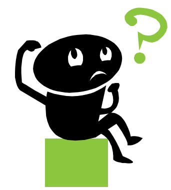
Grafuri euleriene

Istoric. Aplicații

– din cursul 1

Problema celor 7 poduri din Königsberg

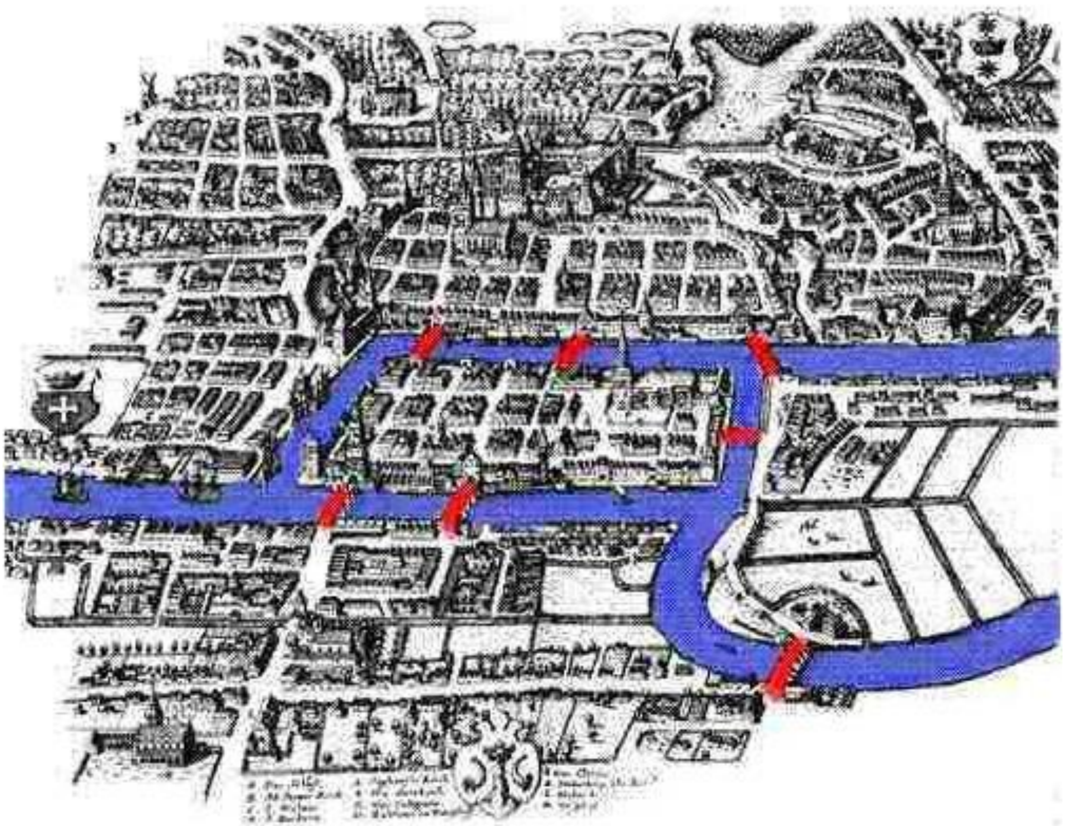


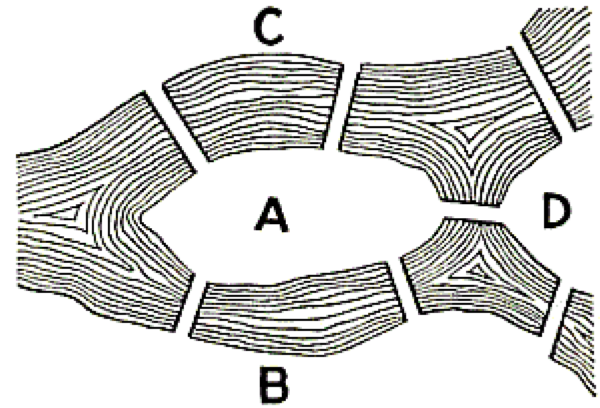


Este posibil ca un om să facă o plimbare în care să treacă pe toate cele 7 poduri o singură dată?

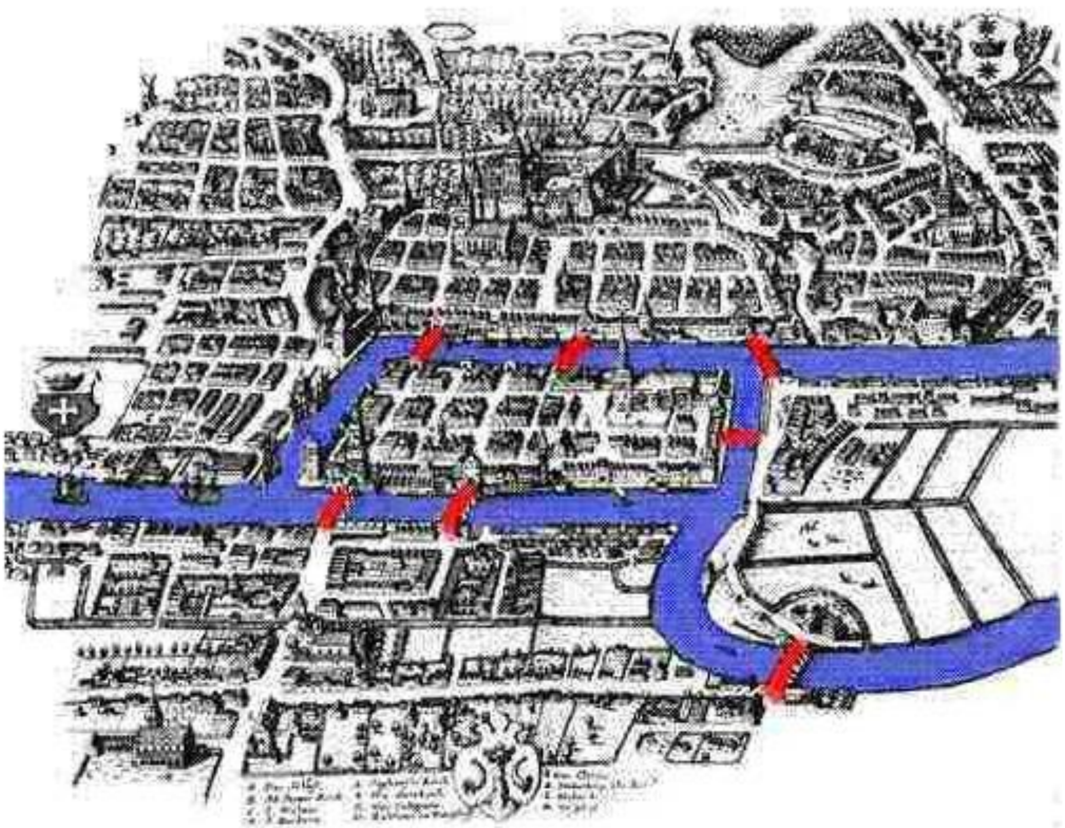
http://think-like-a-git.net/sections/graph-theory/seven-bridges-of-konigsberg.html

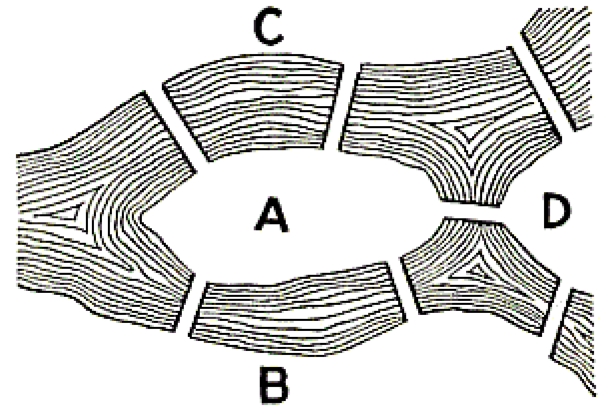
Problema celor 7 poduri din Königsberg



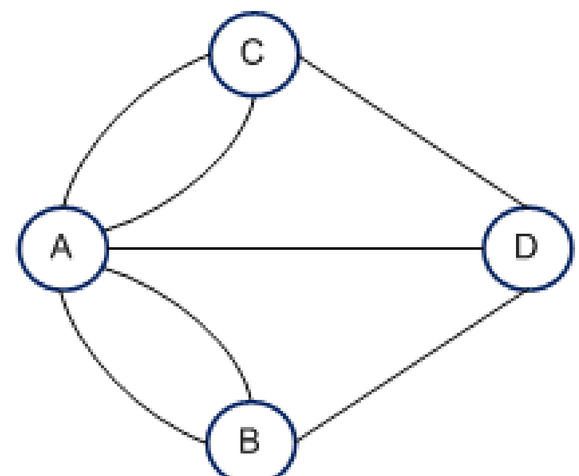


Problema celor 7 poduri din Königsberg

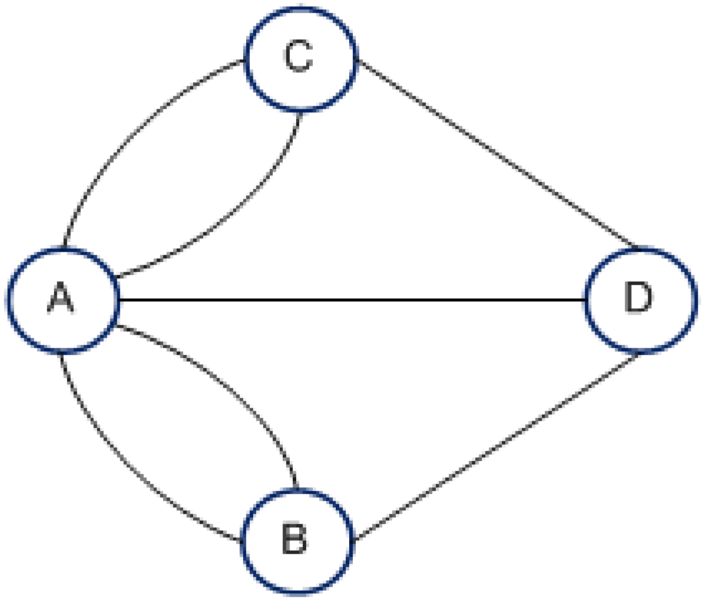




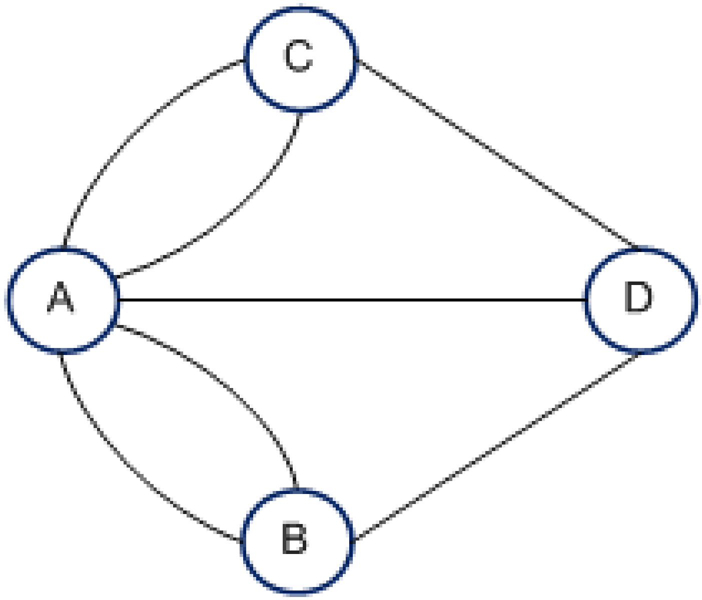
Modelare:

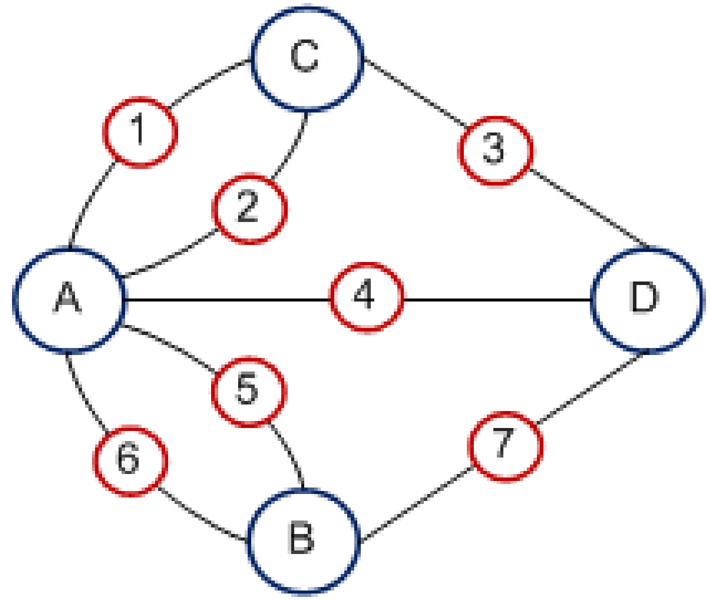


Problema celor 7 poduri din Königsberg



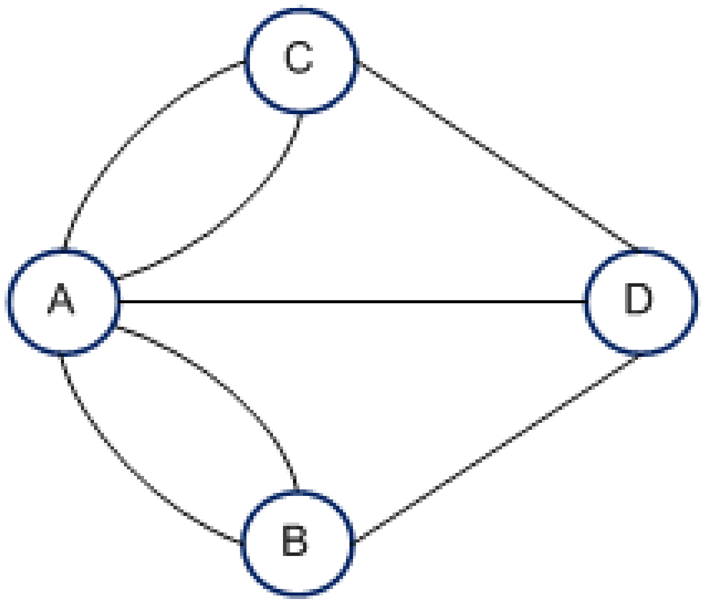
Problema celor 7 poduri din Königsberg

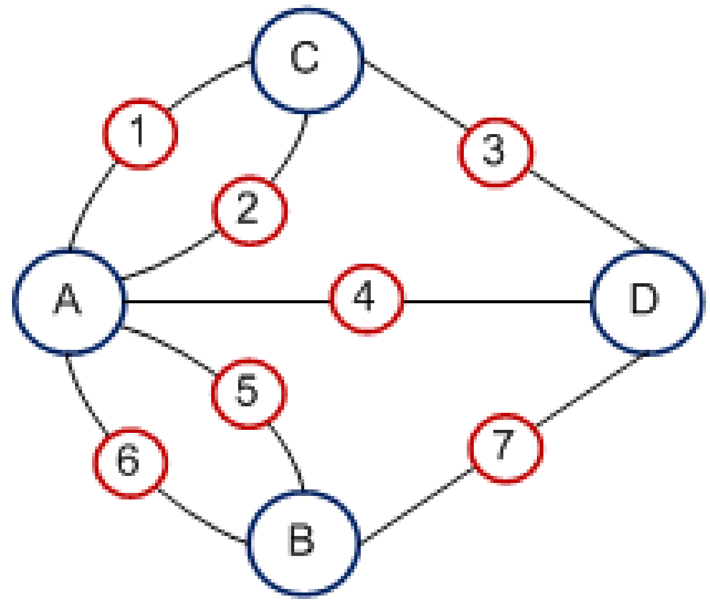




graf simplu

Problema celor 7 poduri din Königsberg





1736 – Leonhard Euler

Solutio problematis ad geometriam situs pertinentis

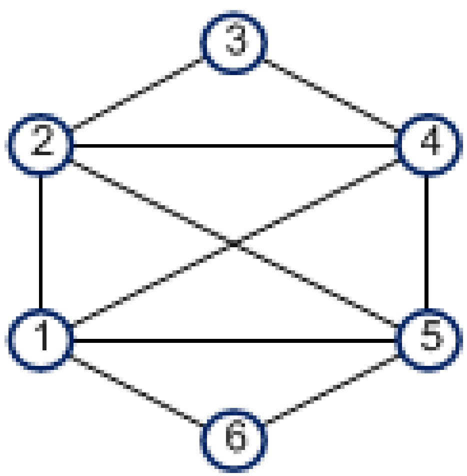
- **Ciclu eulerian** – traseu închis care trece o singură dată prin toate muchiile
- **Graf eulerian**

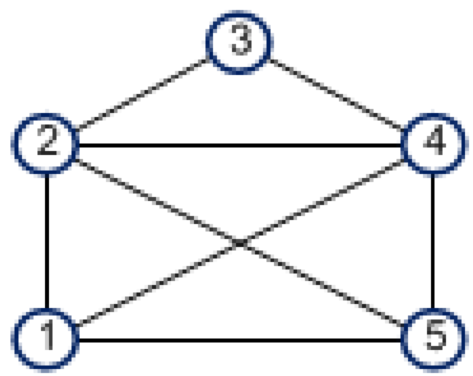
Problema celor 7 poduri din Königsberg

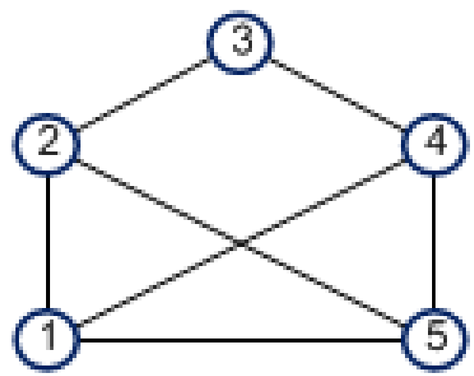
- **Interpretare**

Se poate desena diagrama printr-o curbă continuă închisă fără a ridica creionul de pe hârtie şi fără a desena o linie de două ori (în plus: să terminăm desenul în punctul în care l-am început)?

  - Tăierea unui material



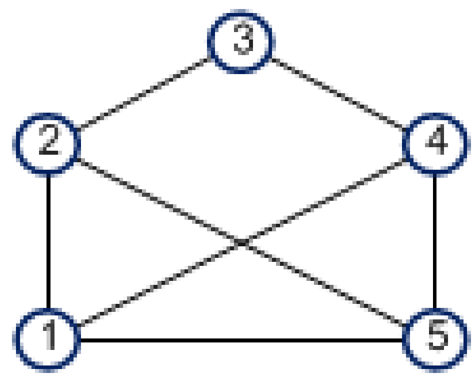




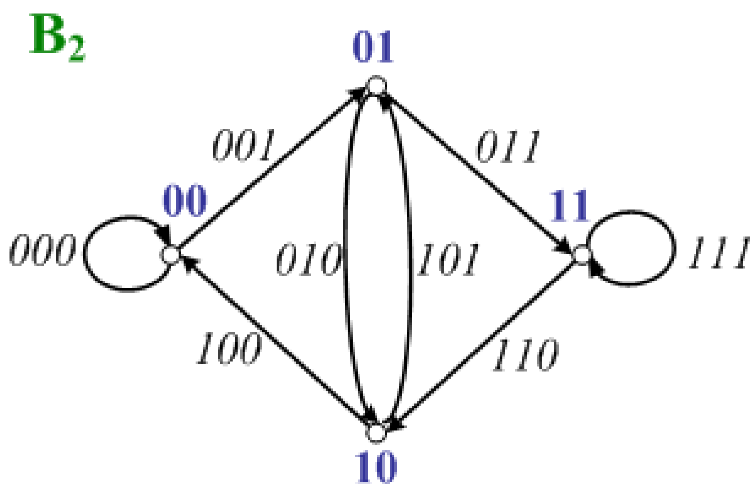
Problema celor 7 poduri din Königsberg

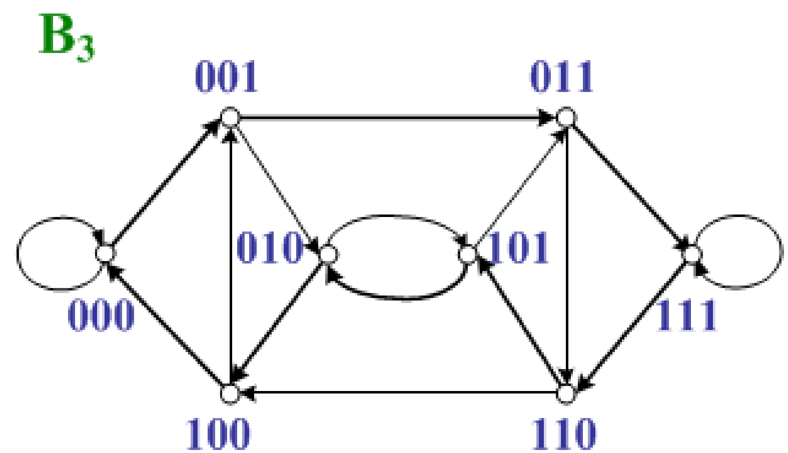
- **Interpretare**

  De câte ori (minim) trebuie să ridicăm creionul de pe hârtie pentru a desena diagrama?

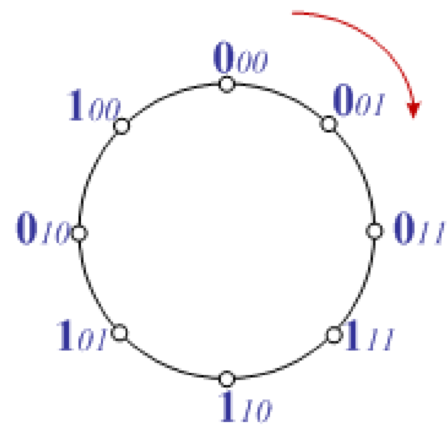


Grafuri de Bruijn





000
001
011
111
110
101
010
100
000



Grafuri euleriene

Fie G graf neorientat

- **Ciclu eulerian** al lui G = ciclu C în G cu

$$E(C)=E(G)$$

- G e**ulerian** = conține un **ciclu** eulerian

- **Lanț eulerian** al lui G = lanț simplu P în G cu

$$E(P) = E(G)$$

Grafuri euleriene

Fie G graf neorientat

- **Ciclu eulerian** al lui G = ciclu C în G cu

$$E(C) = E(G)$$

- G e**ulerian** = conține un ciclu eulerian

- **Lanț eulerian** al lui G = lanț simplu P în G cu

$$E(P) = E(G)$$

Grafuri euleriene

Lemă

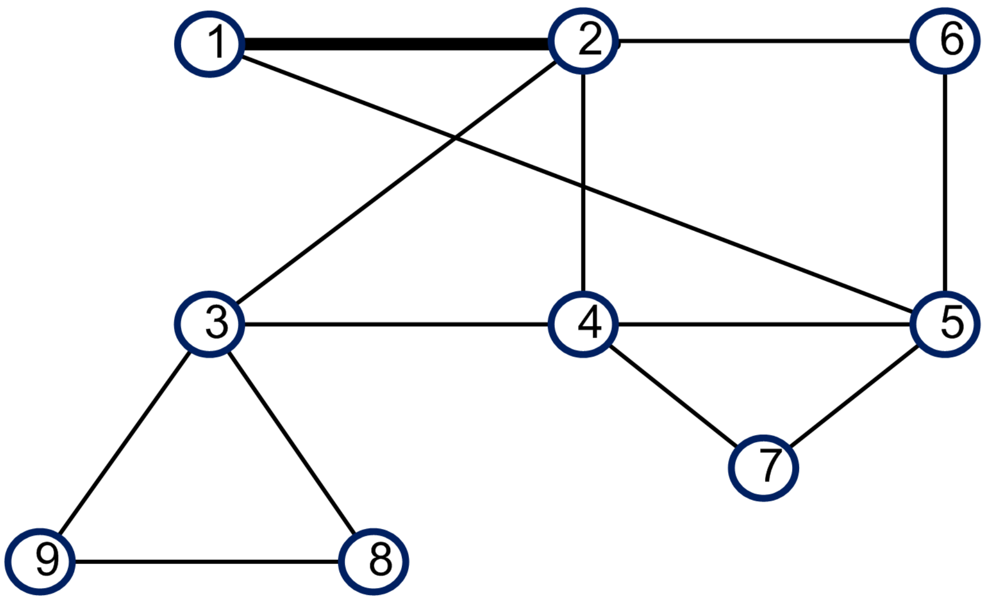
Fie $G=(V,E)$ un graf neorientat, conex, cu **toate vârfurile de grad par** și $E\neq\emptyset$.

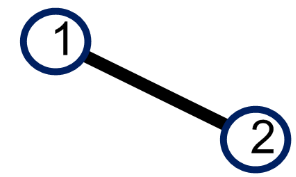
Atunci pentru orice $x \in V$ există un ciclu C în G cu

$x \in V(C)$

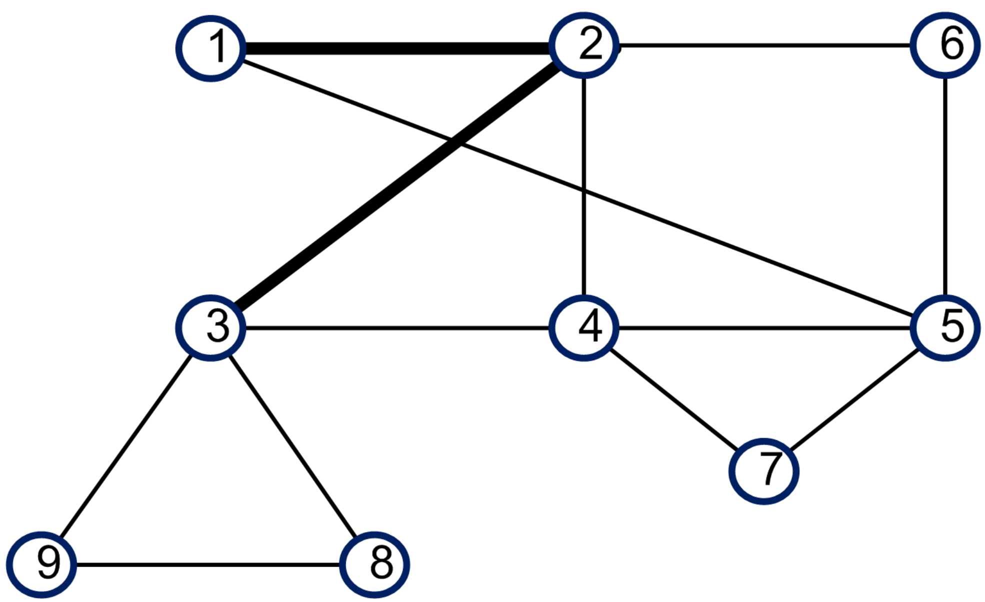
(ciclu care conține x, nu neapărat eulerian)

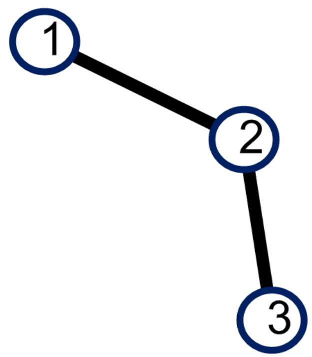
Grafuri euleriene



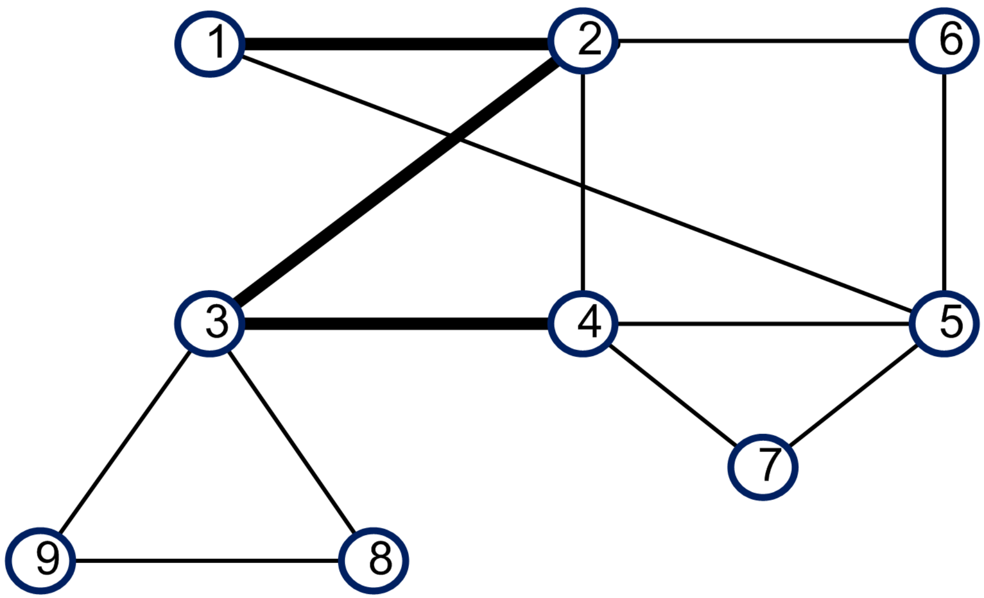


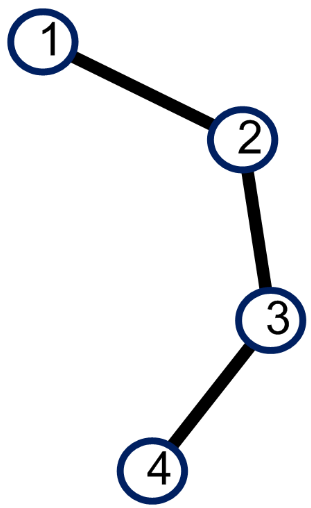
Grafuri euleriene



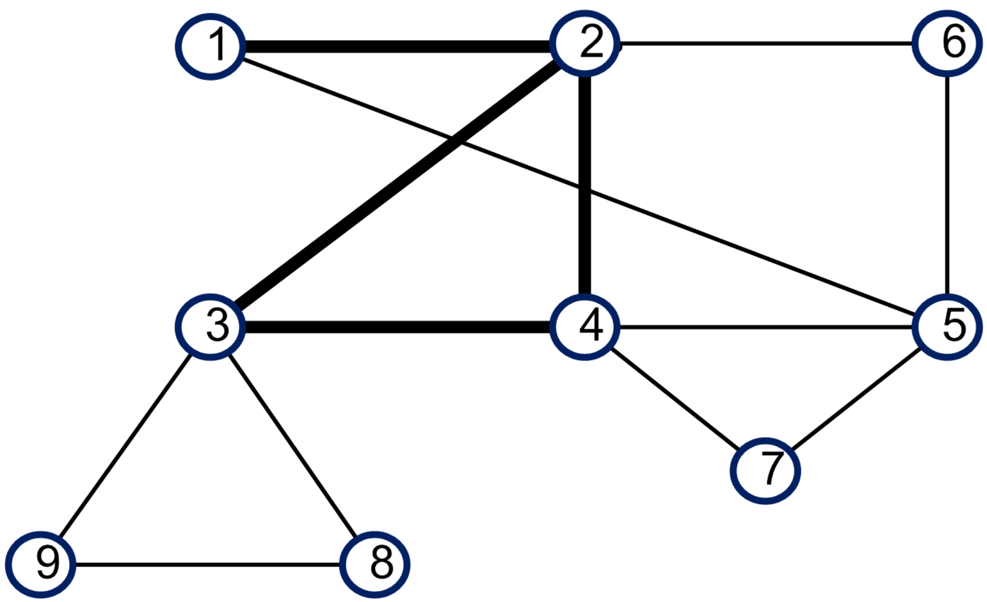


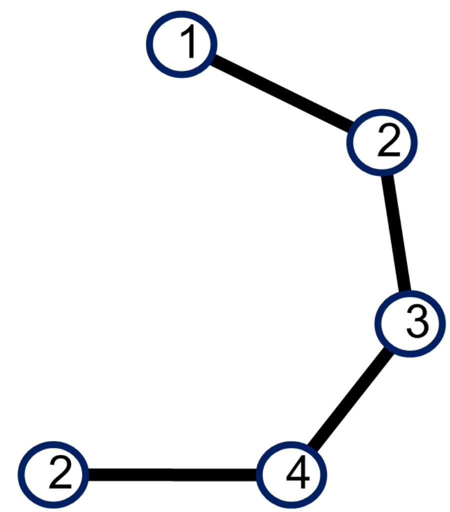
Grafuri euleriene



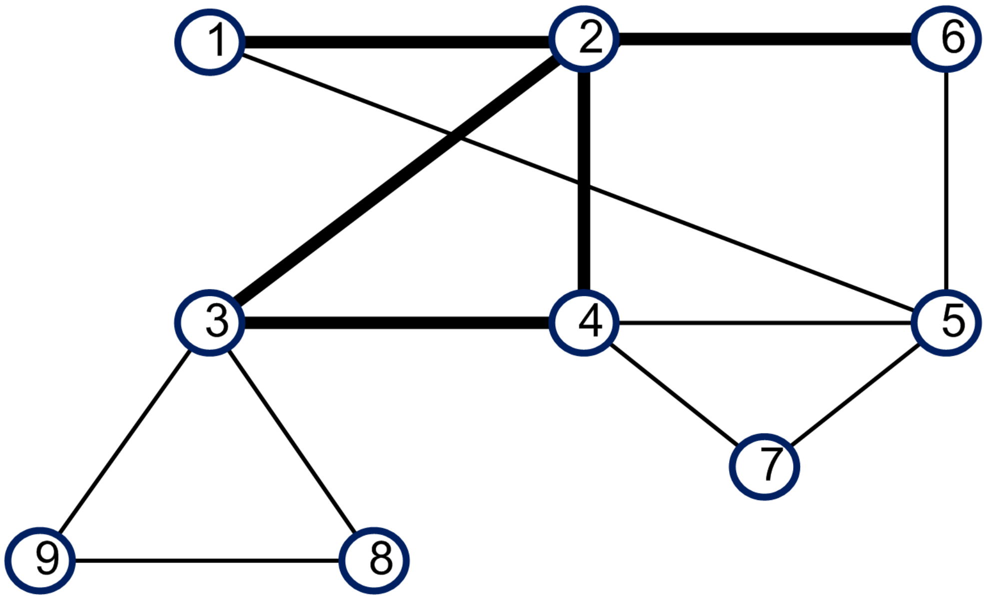


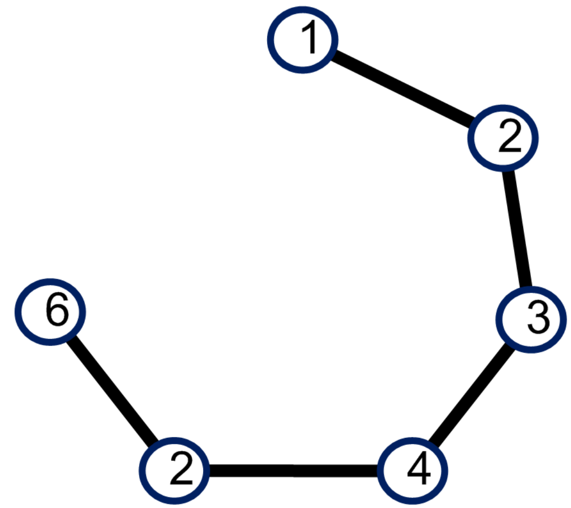
Grafuri euleriene



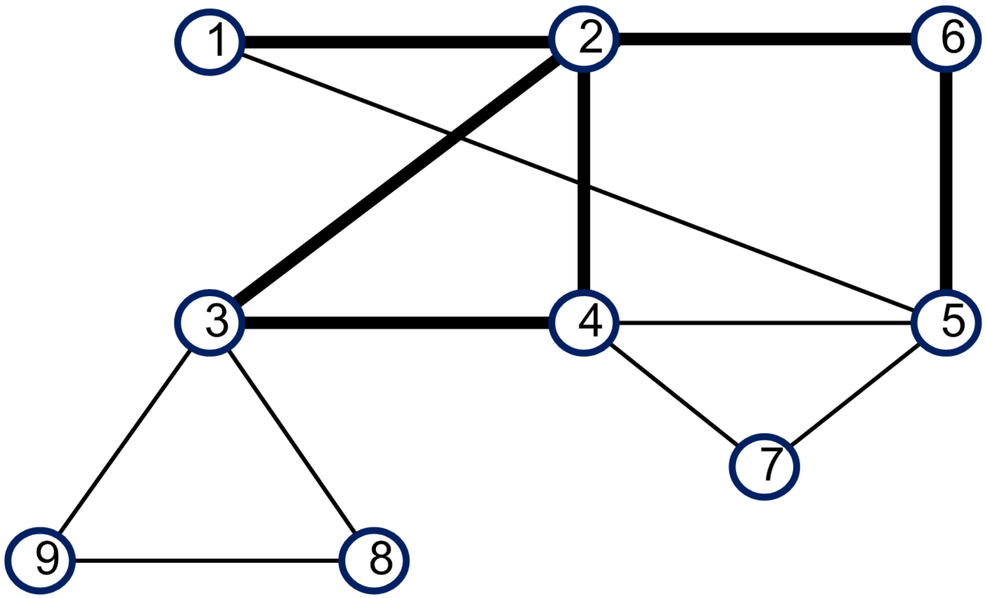


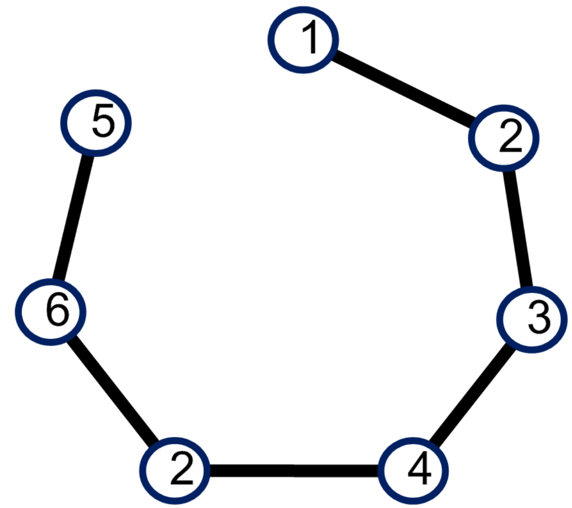
Grafuri euleriene



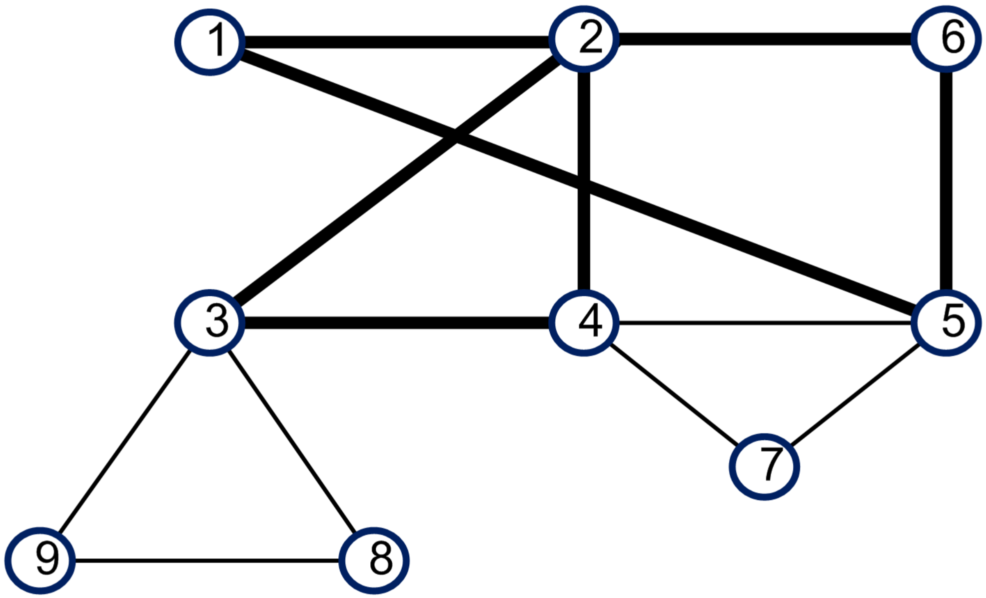


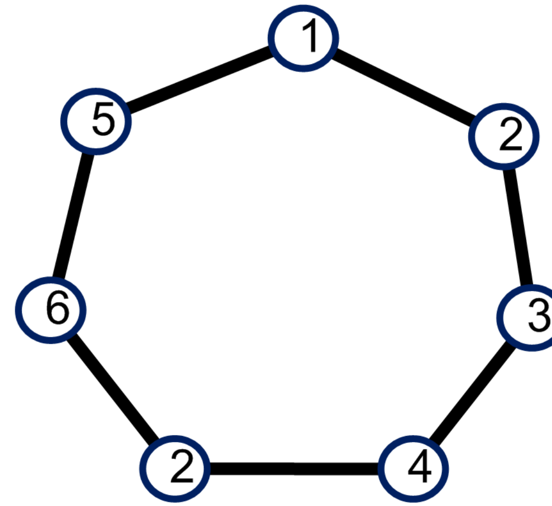
Grafuri euleriene





Grafuri euleriene





Grafuri euleriene

Teorema lui Euler

Fie G=(V, E) un (multi)graf neorientat, conex, cu $E \neq \emptyset$.

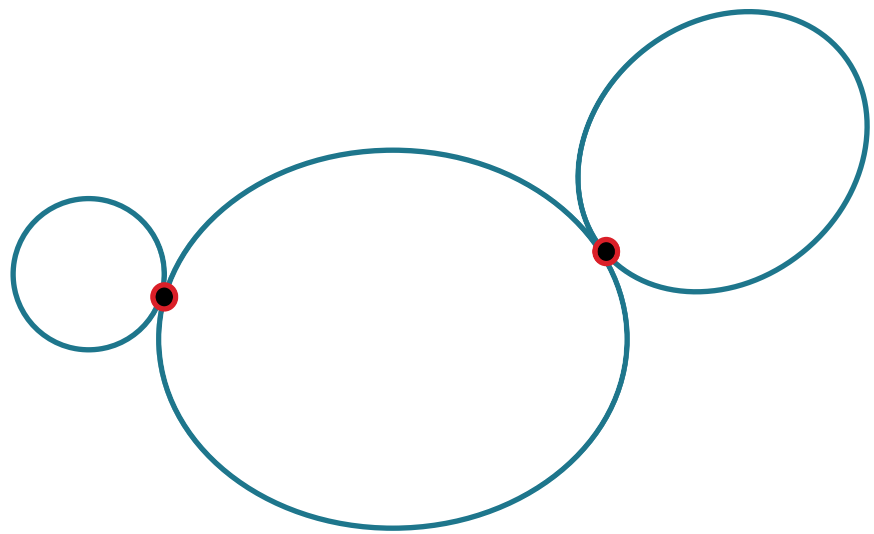
Atunci

G este eulerian $\Leftrightarrow$ **orice vârf din G are grad par**

Algoritmul lui Hierholzer

Determinarea unui ciclu eulerian într-un graf conex (sau un graf conex+ vârfuri izolate) **cu toate vârfurile de grad par**

- bazat pe ideea demonstrației Teoremei lui Euler – **fuziune de cicluri (succesiv)**



Algoritmul lui Hierholzer

- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)

Algoritmul lui Hierholzer

- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- **Pasul 1:**
  - **`alege v ∈ V arbitrar`**
  - **`construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)`**

Algoritmul lui Hierholzer

- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- **Pasul 1:**
  - **`alege v ∈ V arbitrar`**
  - **`construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)`**
- **cât timp $|E(C)| < |E(G)|$ execută**
  - **`selectează v ∈ V(C) cu`** $d_{G-E(C)}(v) > 0$ **`(în care sunt incidente muchii care nu aparțin lui C)`**

Algoritmul lui Hierholzer

- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- **Pasul 1:**
  - **alege v ∈ V arbitrar**
  - **construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)**
- **cât timp |E(C)| < |E(G)| execută**
  - **selectează v ∈ V(C) cu $d_{G-E(C)}$(v) > 0 (în care sunt incidente muchii care nu aparțin lui C)**
  - **construiește C′ un ciclu în G - E(C) care începe cu v**

Algoritmul lui Hierholzer

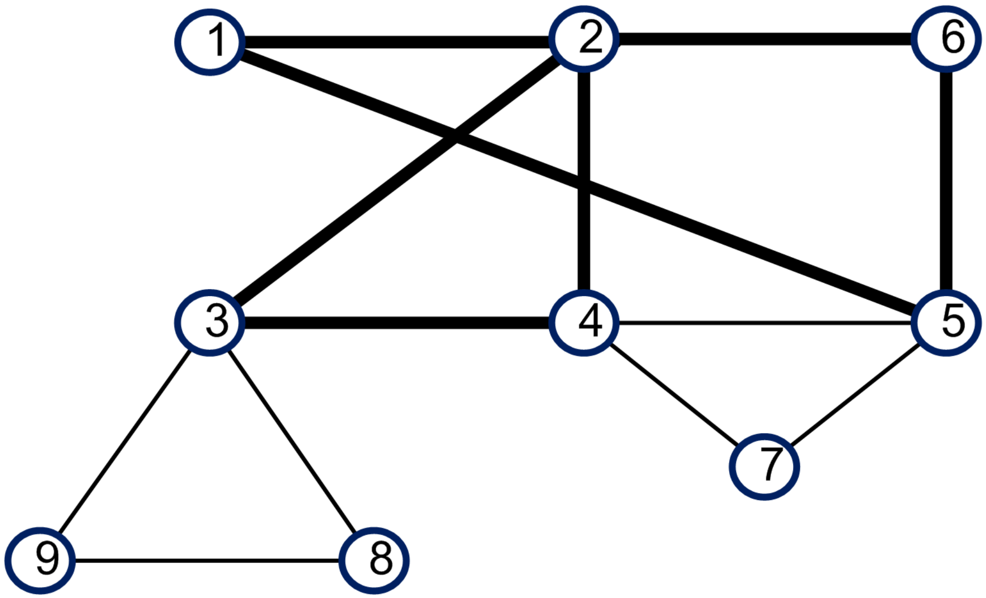
- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- **Pasul 1:**
  - **alege v ∈ V arbitrar**
  - **construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)**
- **cât timp |E(C)| < |E(G)| execută**
  - **selectează v ∈ V(C) cu $d_{G-E(C)}(v) > 0$ (în care sunt incidente muchii care nu aparțin lui C)**
  - **construiește C′ un ciclu în G - E(C) care începe cu v**
  - **C = ciclul obținut prin fuziunea ciclurilor C și C′ în v**

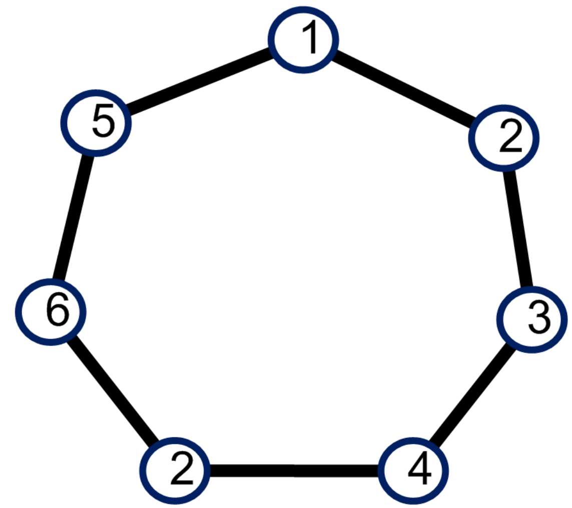
Algoritmul lui Hierholzer

- **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- **Pasul 1:**
  - **alege v ∈ V arbitrar**
  - **construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)**
- **cât timp |E(C)| < |E(G)| execută**
  - **selectează v ∈ V(C) cu $d_{G-E(C)}$(v) > 0 (în care sunt incidente muchii care nu aparțin lui C)**
  - **construiește C′ un ciclu în G - E(C) care începe cu v**
  - **C = ciclul obținut prin fuziunea ciclurilor C și C′ în v**
- **scrie C**

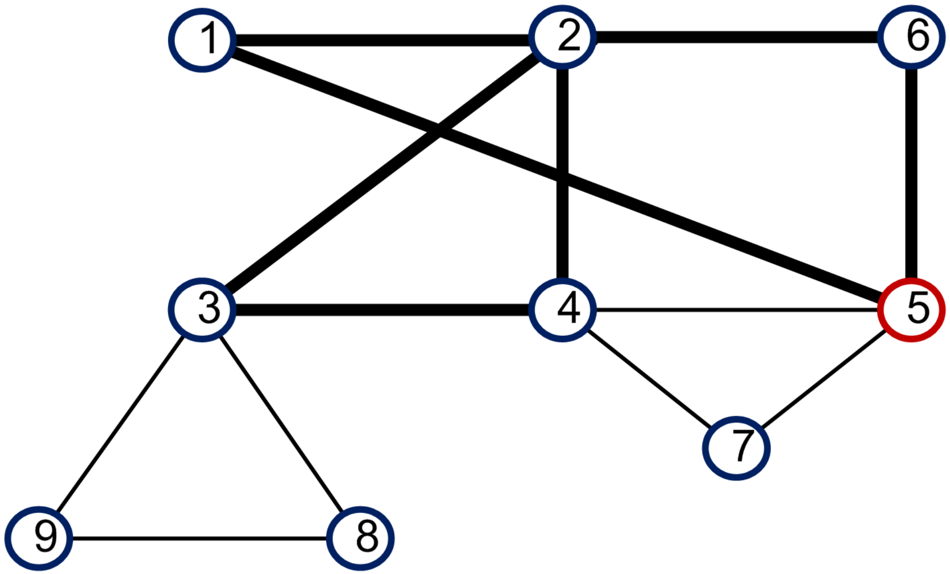
Algoritmul lui Hierholzer

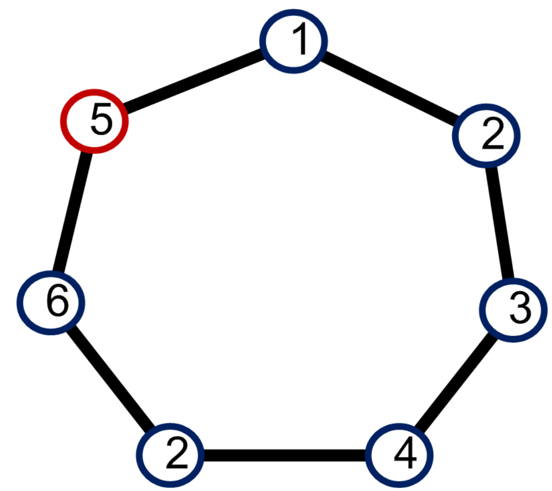
Pornim cu ciclul construit cu algortimul din Lema 1





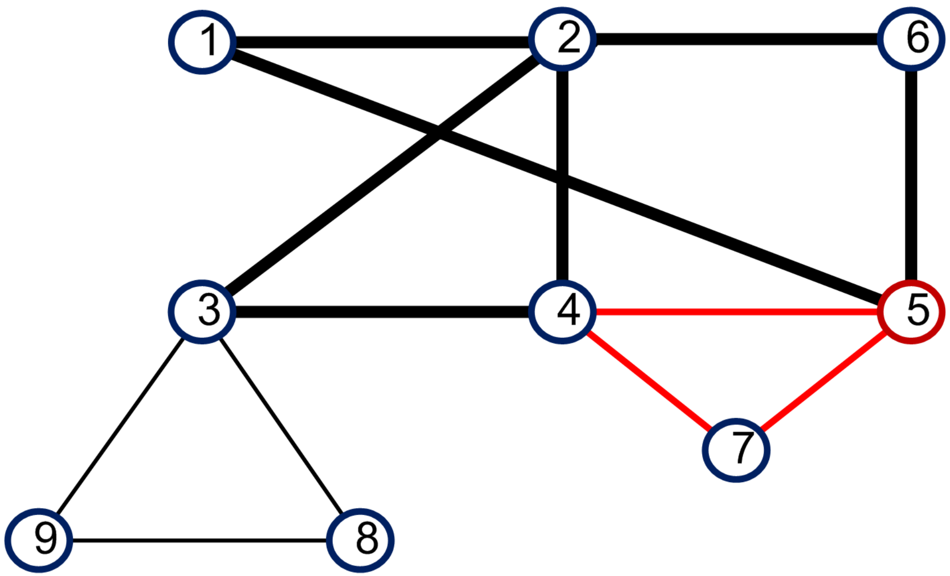
Algoritmul lui Hierholzer

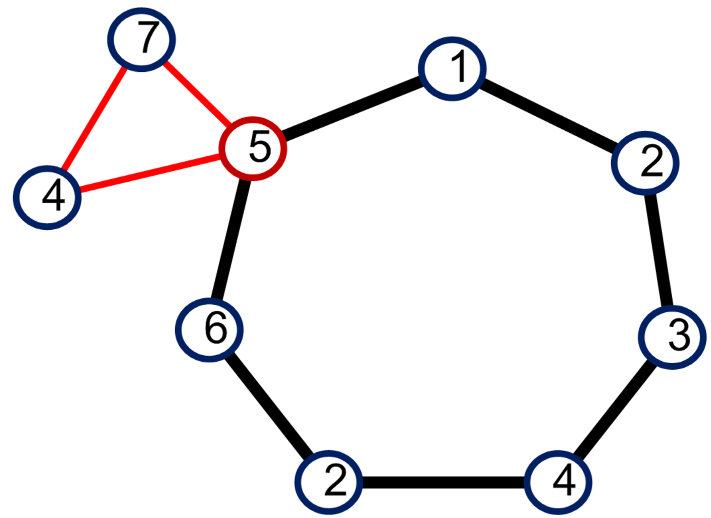




$C_1 = [1, 2, 3, 4, 2, 6, 5, 1]$

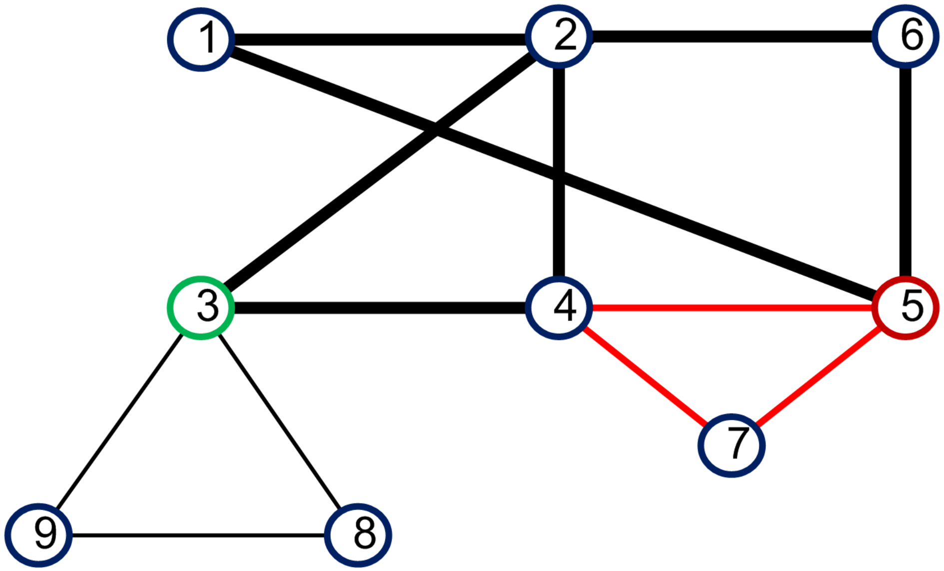
Algoritmul lui Hierholzer

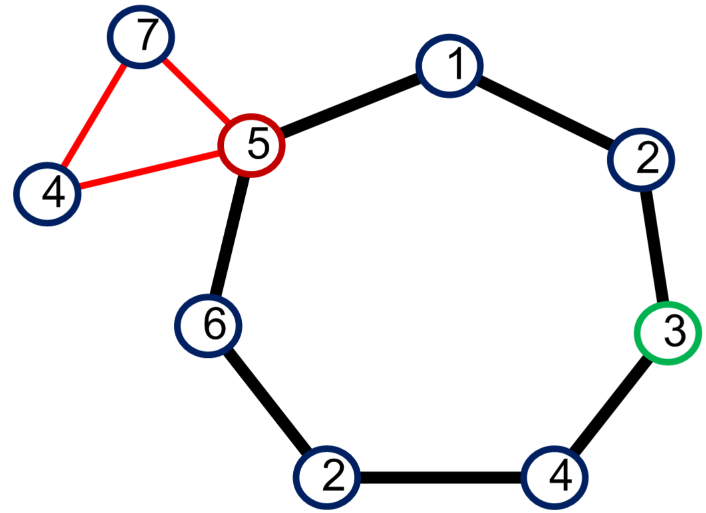




$C_2 = [1, 2, 3, 4, 2, 6, 5, 4, 7, 5, 1]$

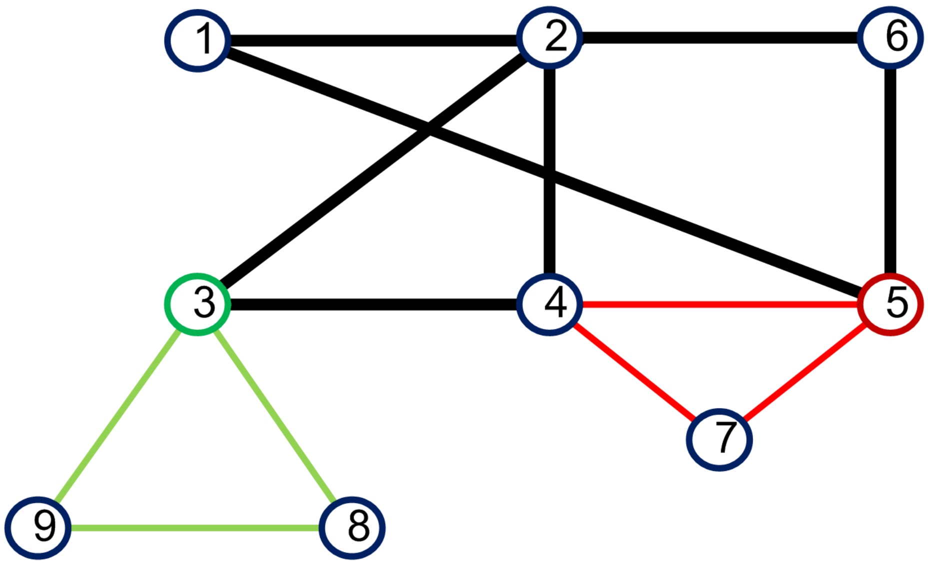
Algoritmul lui Hierholzer

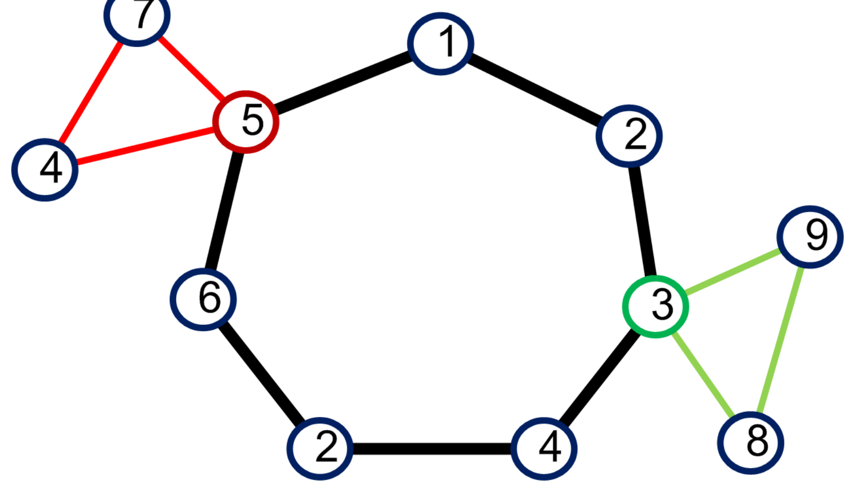




$C_2 = [1, 2, \textcolor{green}{3}, 4, 2, 6, 5, 4, 7, 5, 1]$

Algoritmul lui Hierholzer





$$C_3 = [1, 2, 3, 8, 9, 3, 4, 2, 6, 5, 4, 7, 5, 1]$$

Algoritmul lui Hierholzer

- **Complexitate – O(m)**

Algoritmul lui Hierholzer

- **Posibilă implementare**
  - **Varianta 1**– Liste dublu înlănțuite/stive
  - Muchiile folosite - marcate (nu neapărat șterse)

Algoritmul lui Hierholzer

- Varianta 2 – posibilă implementare recursivă

```
euler (nod v)
    cat timp d(v)> 0
          alege vw o muchie incidenta in v
          sterge muchia vw din G
          euler (w)
    C = C + v  //adaugam v la ciclul C
```

Inițial

```
C = Ø
euler(1)  //pornim construcția din varful 1
```

Algoritmul lui Hierholzer

- **Varianta 2 – posibilă implementare recursivă**

```
euler (nod v)
    cat timp d(v)> 0
        alege vw o muchie incidenta in v
        sterge muchia vw din G
        euler (w)
    C = C + v  //adaugam v la ciclul C
```

Observație – putem alege muchiile incidente în v, de exemplu, în ordinea dată de listele de adiacență

```
cat timp d(v)> 0
  alege vw o muchie incidenta in v
  sterge muchia vw din G
  euler (w)
```

⟹

```
pentru vw∈ E
  sterge muchia vw din G
  euler (w)
```

Algoritmul lui Hierholzer

- Varianta 2 – posibilă implementare recursivă

```
euler (nod v)
    cat timp d(v)> 0
          alege vw o muchie incidenta in v
          sterge muchia vw din G
          euler (w)
    C = C + v  //adaugam v la ciclul C
```

Inițial

```
C = Ø
euler(1)  //pornim construcția din varful 1
```

- http://www.infoarena.ro/problema/ciclueuler

Lanțuri euleriene

Teorema lui Euler

Fie $G=(V, E)$ un graf neorientat, conex, cu $E \neq \emptyset$.

Atunci

G are un lanț eulerian $\Leftrightarrow$ **G are cel mult două vârfuri de grad impar**

Descompuneri euleriene în lanțuri

- **k-descompunere euleriană în lanțuri** a unui graf G =

  o mulțime de k lanțuri simple, muchie-disjuncte

  $\Delta = \{P_1, P_2, \ldots, P_k\}$

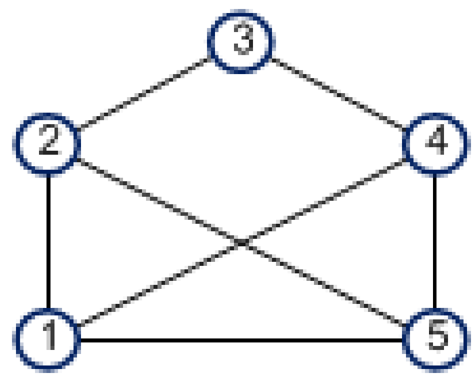
  ale căror muchii induc o k-partiție a lui E(G):

  $E(G) = E(P_1) \cup E(P_2) \cup \ldots \cup E(P_k)$

Descompuneri euleriene în lanțuri

- **Interpretare**

  De câte ori (minim) trebuie să ridicăm creionul de pe hârtie pentru a desena diagrama?



Descompuneri euleriene în lanțuri

Teoremă – Descompunere euleriană

Fie G=(V, E) un graf orientat, conex (= graful neorientat asociat este conex), cu **exact 2k vârfuri de grad impar** ($k>0$). Atunci există o k-descompunere euleriană a lui G și k este cel mai mic cu această proprietate.

Grafuri orientate euleriene

Grafuri orientate euleriene

Teorema lui Euler

Fie G=(V, E) un graf orientat, conex (= graful neorientat asociat este conex), cu $E \neq \emptyset$.

Atunci

G este eulerian $\Leftrightarrow \forall\, v \in V \quad d_G^-(v) = d_G^+(v)$

Lanțuri euleriene

Teorema lui Euler

Fie G=(V, E) un (multi)graf neorientat, conex, cu E≠∅.

Atunci

G are un drum eulerian ⇔

(∀ v∈V $d_G^-(v) = d_G^+(v)$) **sau**

(∃ x∈V cu $d_G^-(x) = d_G^+(x) - 1$,

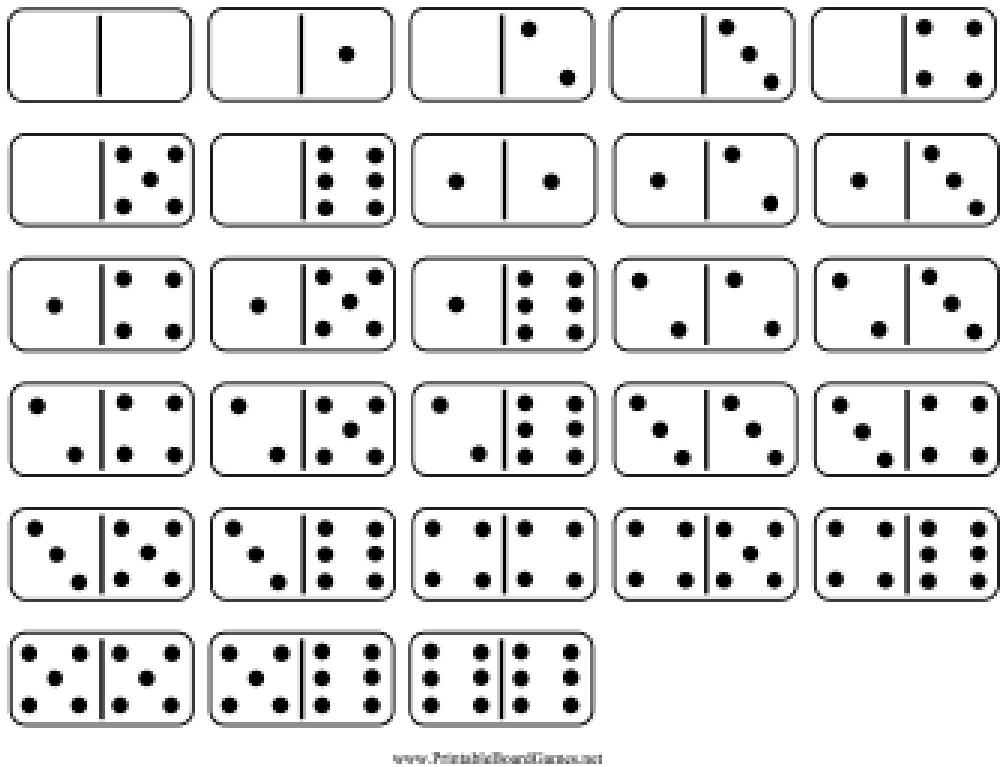
∃ y∈V cu $d_G^-(y) = d_G^+(y) + 1$,

∀ v∈V−{x, y} $d_G^-(v) = d_G^+(v)$)

Grafuri euleriene

Problemă – joc domino

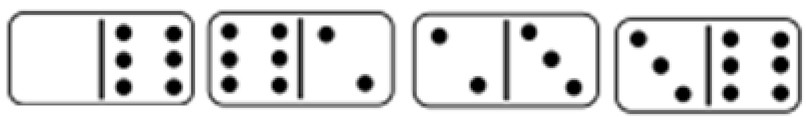
Piesă de domino – două fețe, numere 0..6



Grafuri euleriene

Problemă – joc domino

Șir de piese de domino – respectă regula de construcție: primul număr de pe piesa adăugată la șir = al doilea număr de pe ultima piesă din șir



Grafuri euleriene

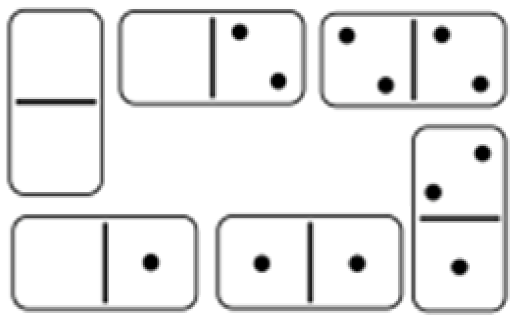
Problemă – joc domino

Se poate forma un șir de piese de domino care să conțină toate piesele + să se termine cu același număr cu care a început (un șir circular)?

Grafuri euleriene

Problemă – joc domino

Exemplu – daca folosim doar piese cu numere 0..2 putem forma un ciclu



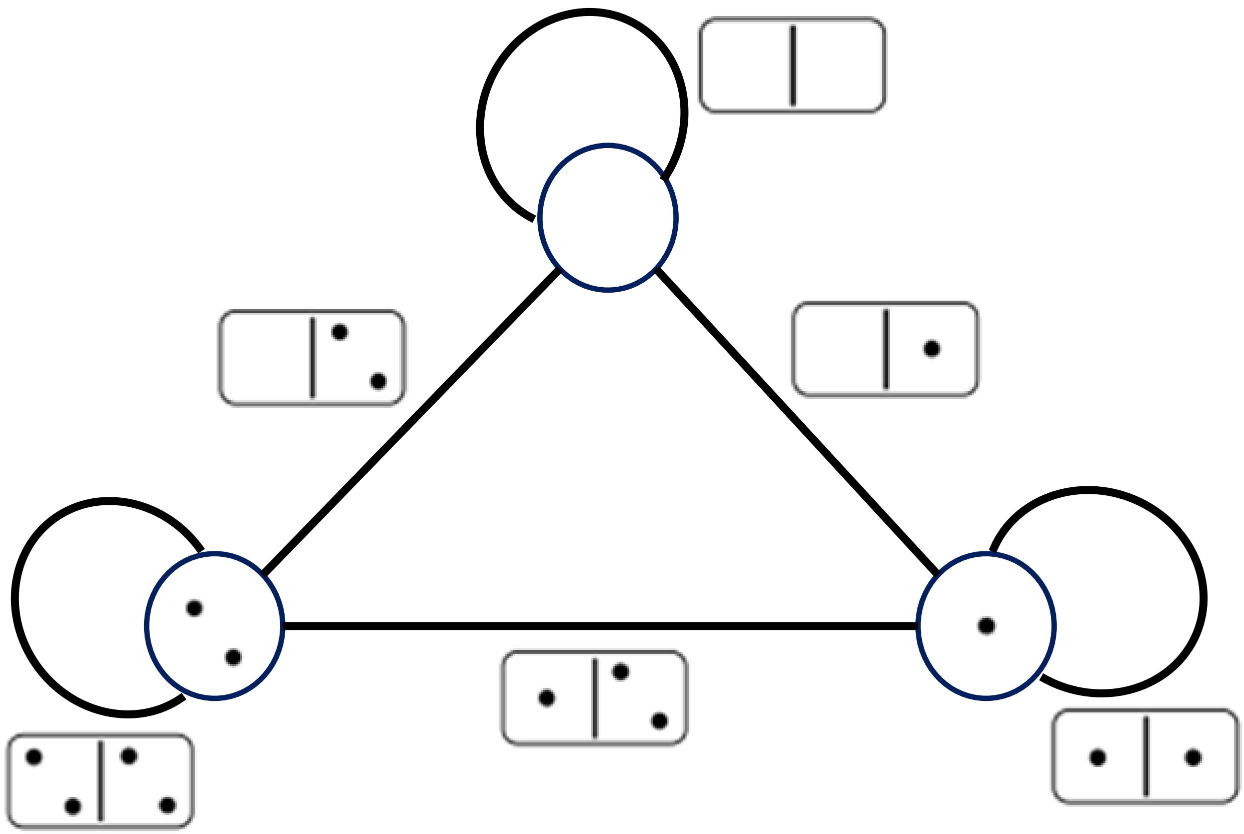
Grafuri euleriene

Problemă – joc domino

Graf asociat

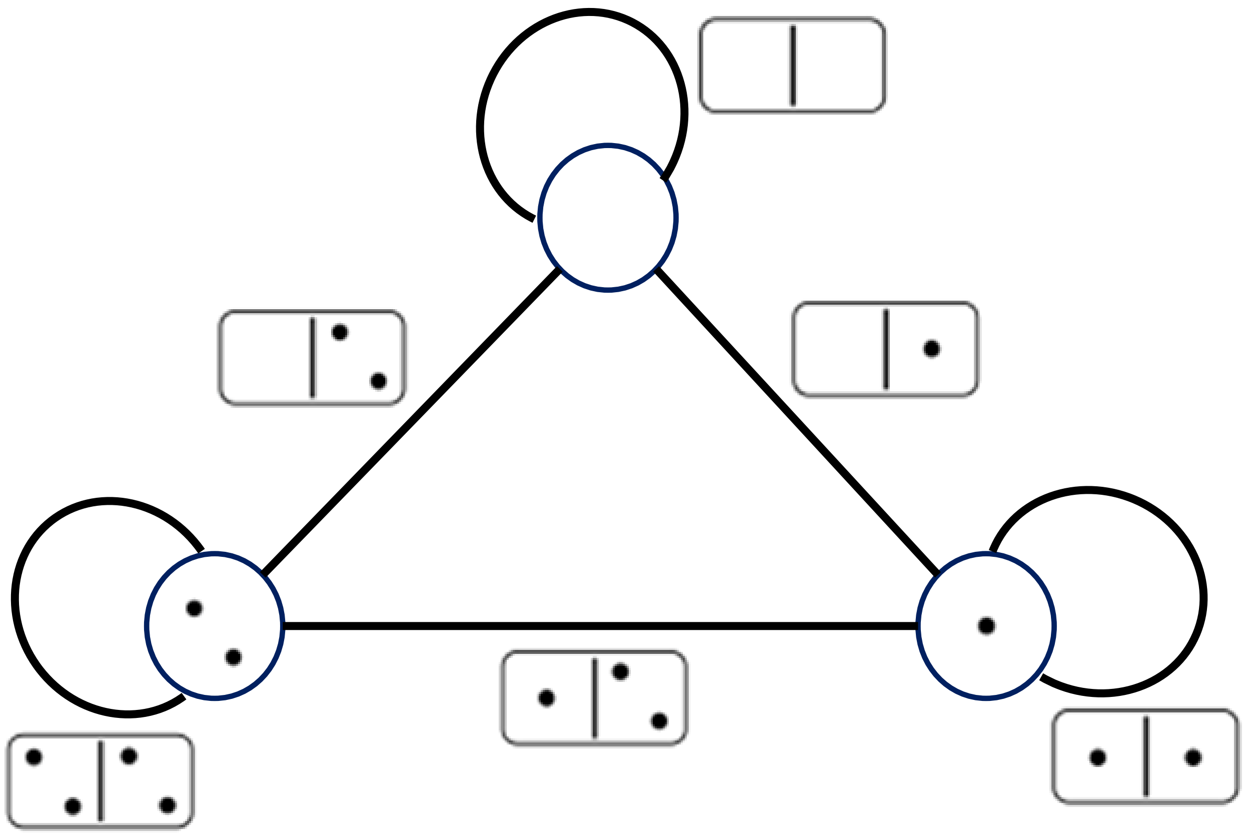
Grafuri euleriene

Problemă – joc domino



Grafuri euleriene

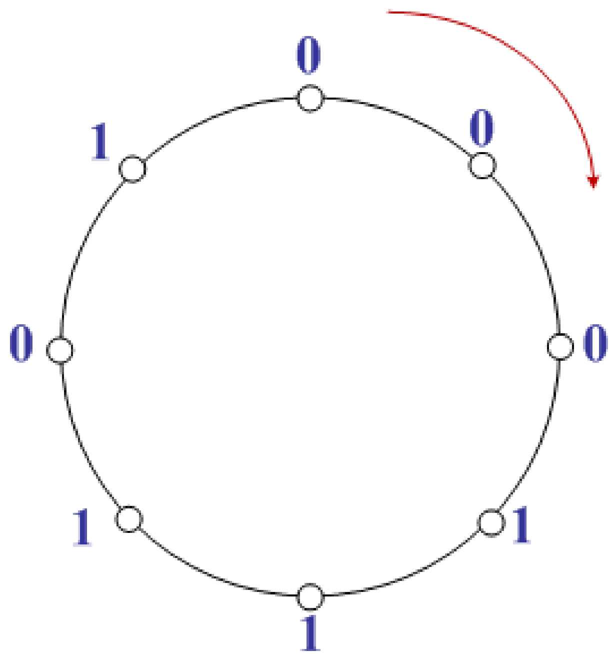
Există ciclu de piese ⇔ există ciclu eulerian în (multi)graf

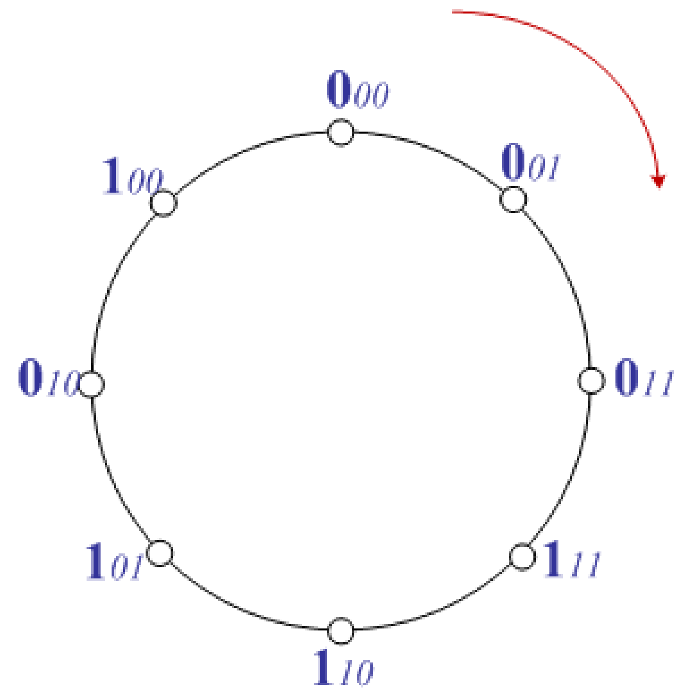


Grafuri euleriene

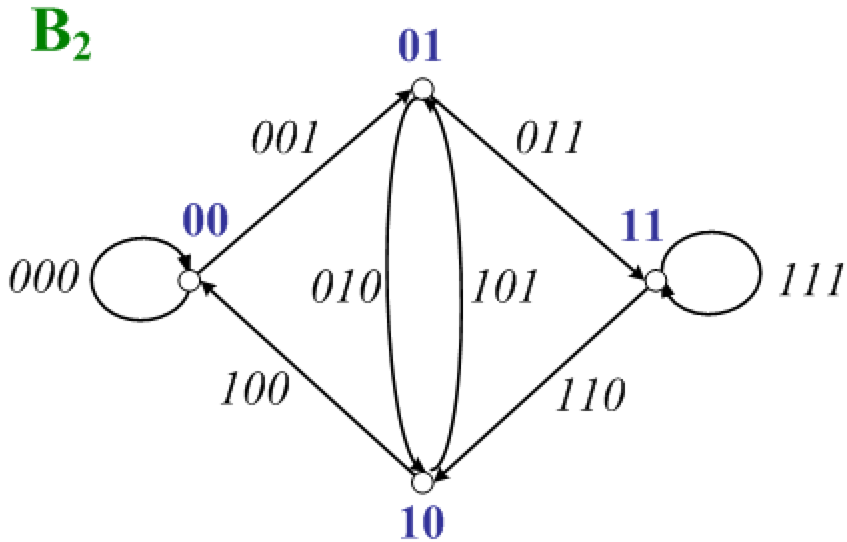
Problema lui POSTHUMUS

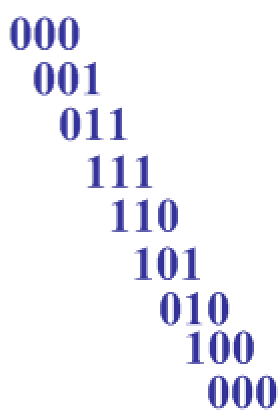
- f (n) = numărul minim de cifre de 0 și 1 care se pot dispune circular a.î. între cele f (n) secvențe de lungime n de cifre succesive apar toți cei 2^n vectori de lungime n peste {0,1} (citite în același sens).
- Evident $f(n) \geq 2^n$.

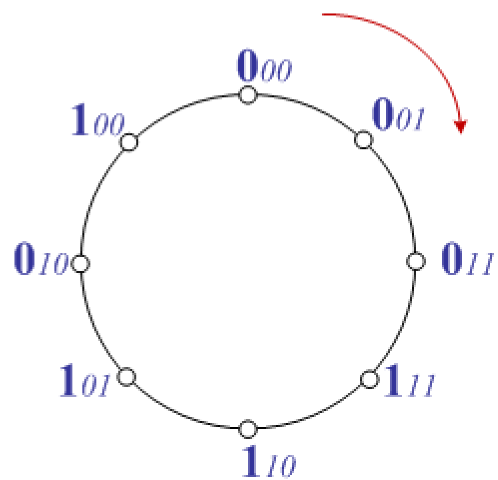




Grafuri de Bruijn







Grafuri de Bruijn

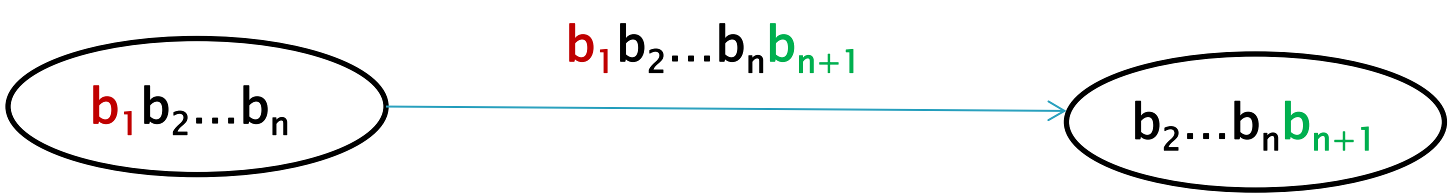
- Multigraf
- $V(B_n) = \{0,1\}^n$ (mai general $\{0,1,\dots,\ p\}^n$)

  (sau cuvinte de lungime n peste un alfabet finit)

- $E(B_n)$ etichetate cu $\{0,1\}^{n+1}$ ($\{0,1,\dots,\ p\}^{n+1}$)

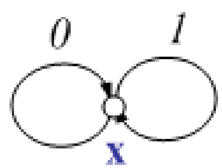
  $b_1 b_2 \dots b_n b_{n+1}$ etichetează arcul de la

  $b_1 \mathbf{b_2 \dots b_n}$ la $\mathbf{b_2 \dots b_n} b_{n+1}$

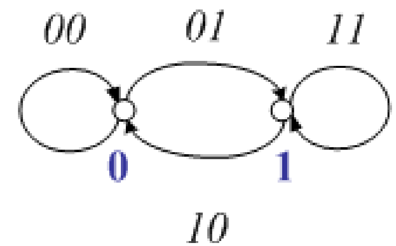


Grafuri de Bruijn

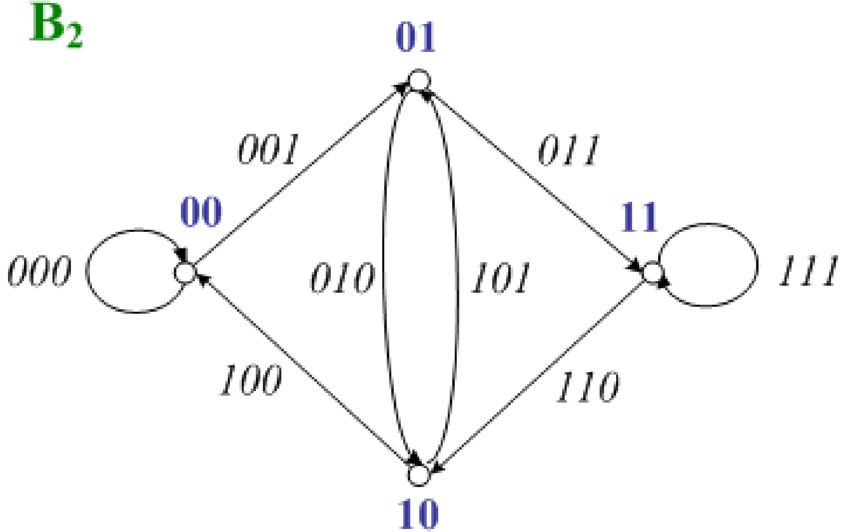
B_0



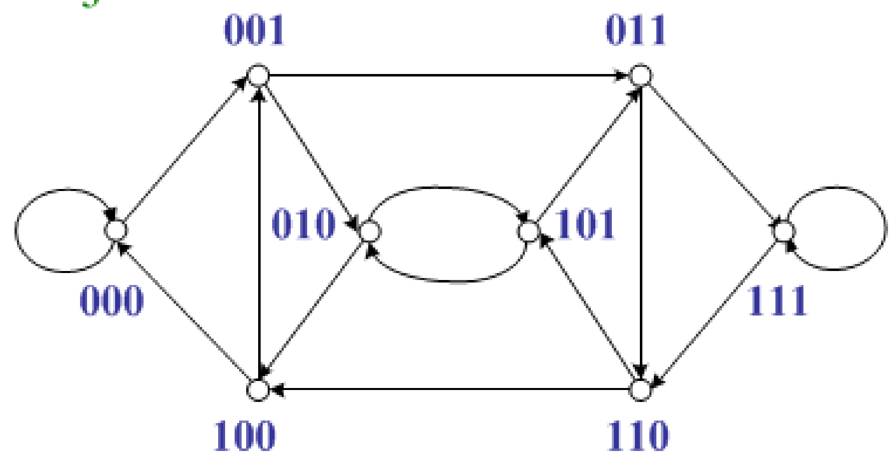
B_1



B_2



B_3



Grafuri de Bruijn

- B_n este eulerian

  $d^+(v) = ?$

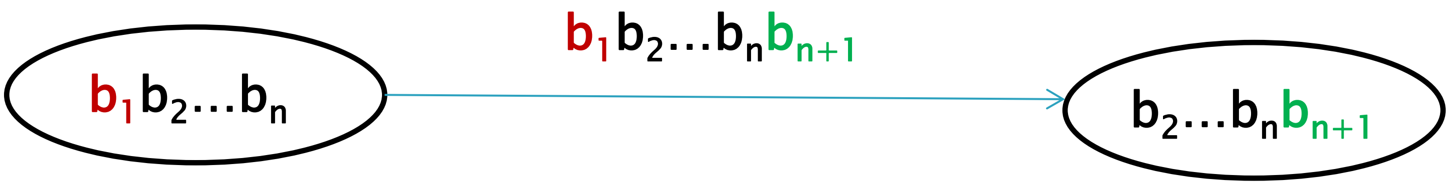
  $d^-(v) = ?$

Grafuri de Bruijn

- B_n este eulerian

  $d^+(v) = ?$

  $d^-(v) = ?$



v

orice b_{n+1} din alfabet

Grafuri de Bruijn

- B_n este eulerian

  $d^+(v) = ?$

  $d^-(v) = ?$

- **Prima cifră din etichetele arcelor unui circuit eulerian în B_{n-1} – soluție pentru problema lui Posthumus**

Grafuri de Bruijn

- B_n este eulerian

  $d^+(v) = ?$

  $d^-(v) = ?$

- **Prima cifră din etichetele arcelor unui circuit eulerian în B_{n-1} – soluție pentru problema lui Posthumus**

- **Observație**

  Circuit eulerian in $B_{n-1} \leftrightarrow$ circuit hamiltonian in B_n

Grafuri de Bruijn

- Aplicație – genetică *(Genome Assembly)*