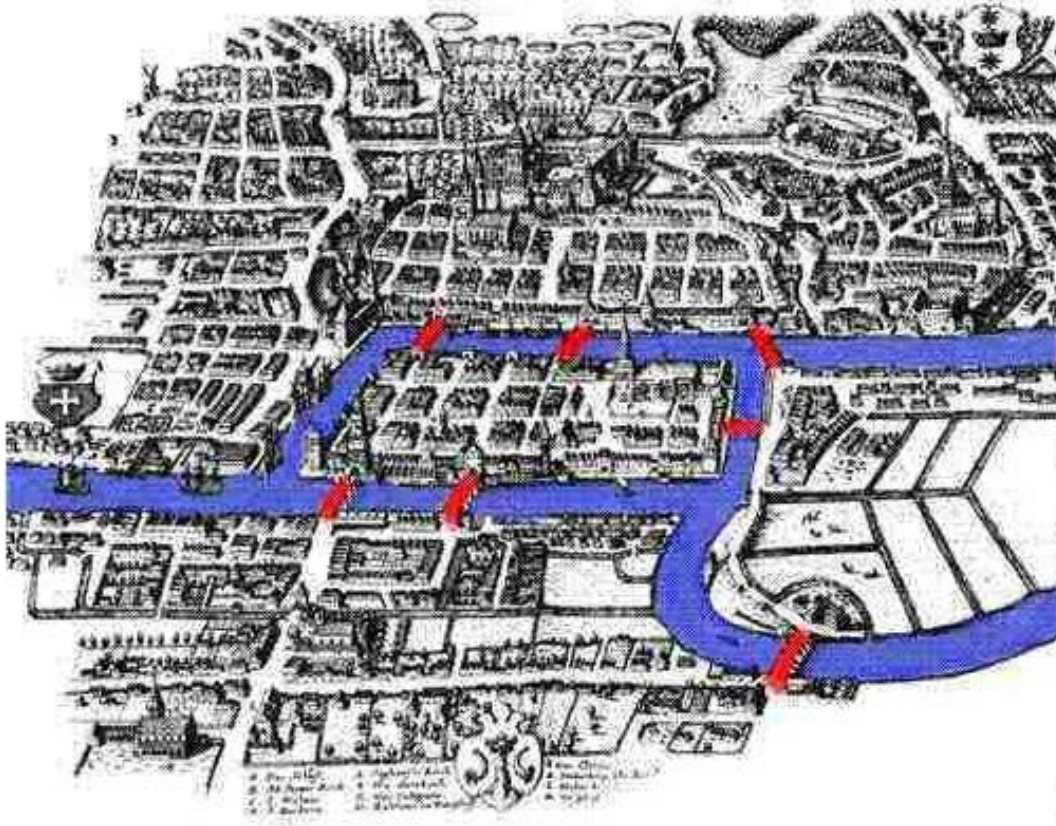


Grafuri euleriene

Istoric. Aplicații

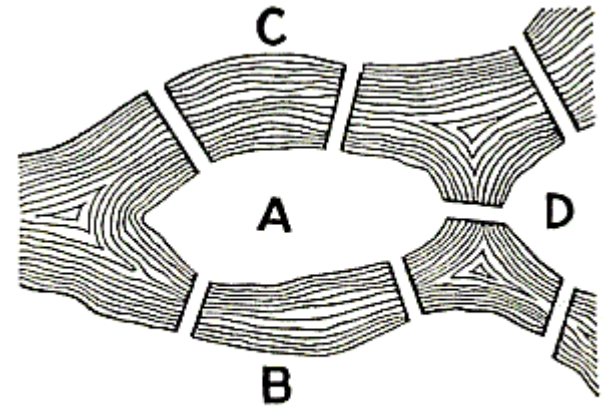
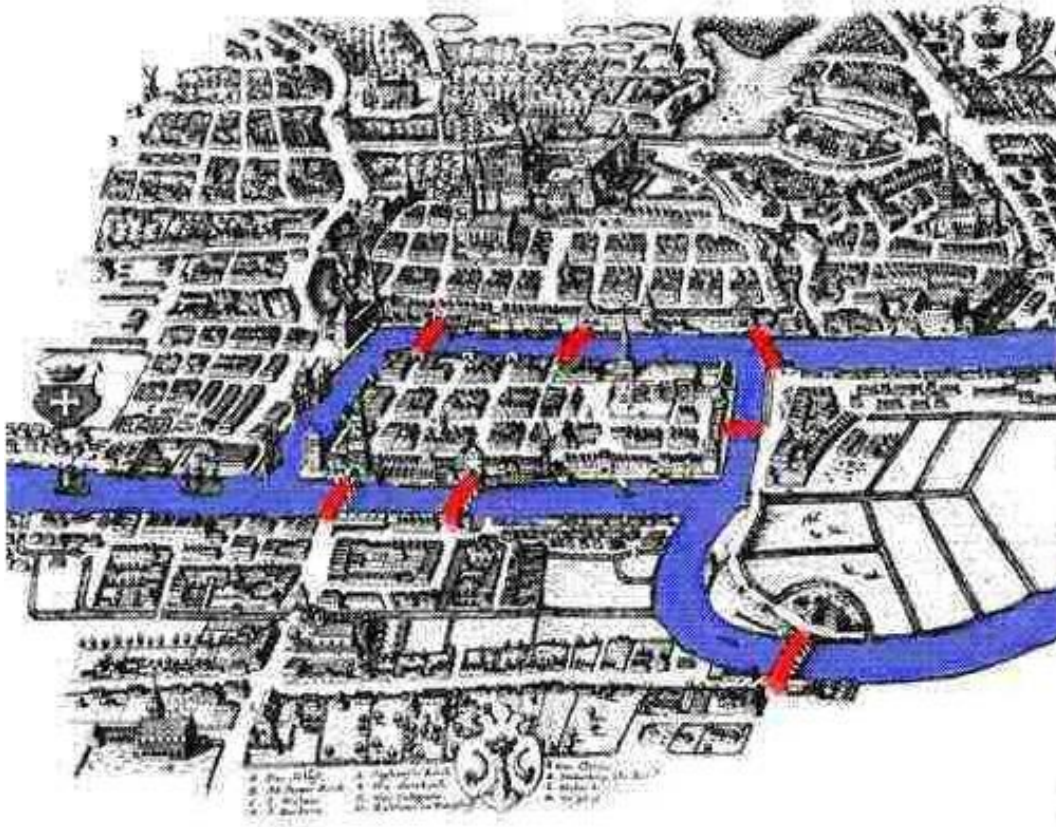
– din cursul 1

Problema celor 7 poduri din Königsberg

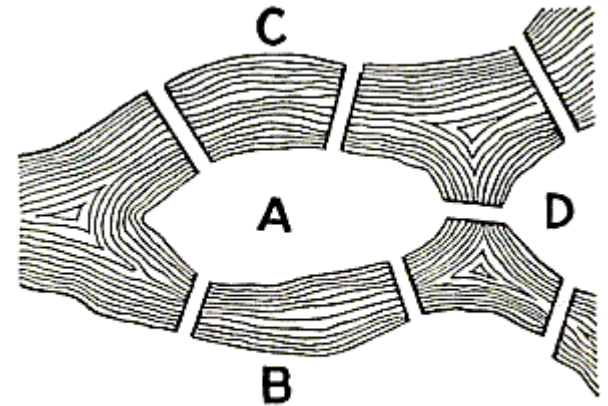
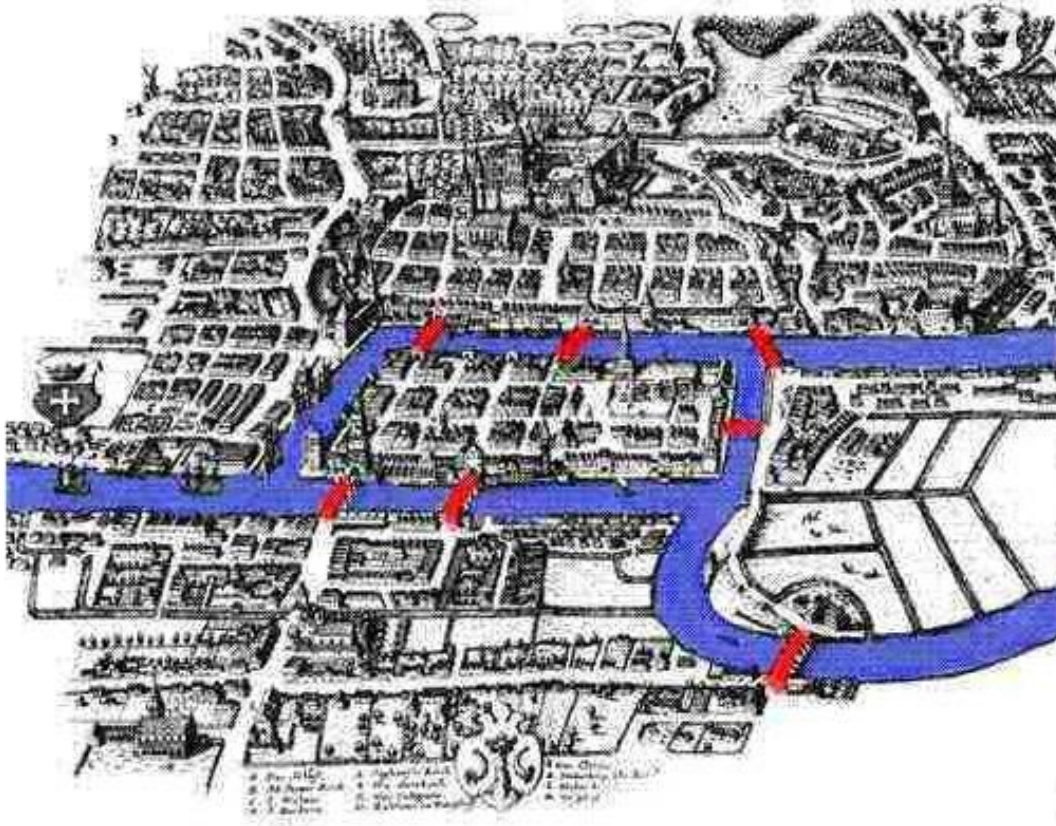


Este posibil ca un om să facă o plimbare în care să treacă pe toate cele 7 poduri o singură dată?

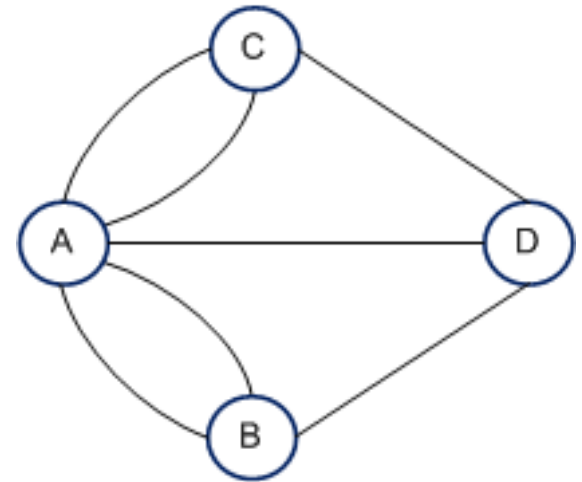
Problema celor 7 poduri din Königsberg



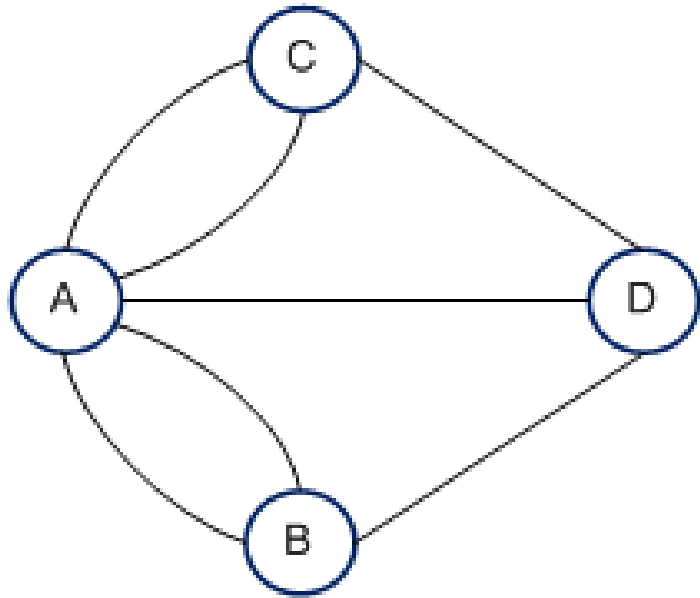
Problema celor 7 poduri din Königsberg



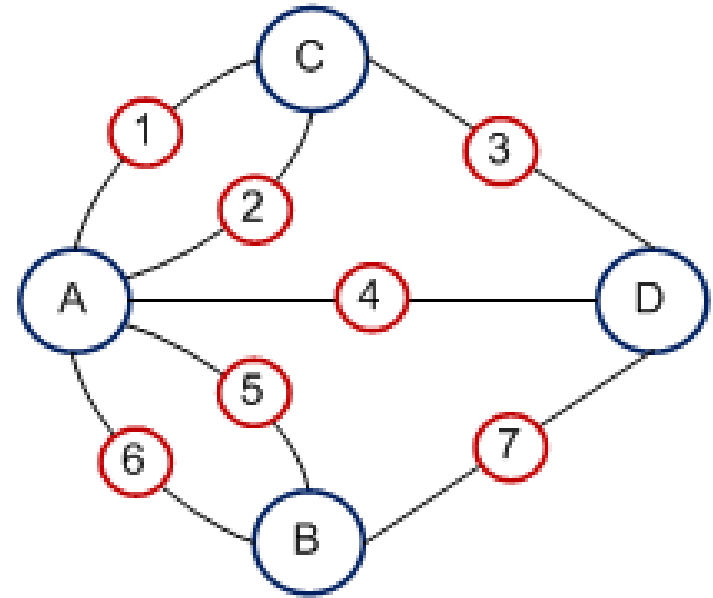
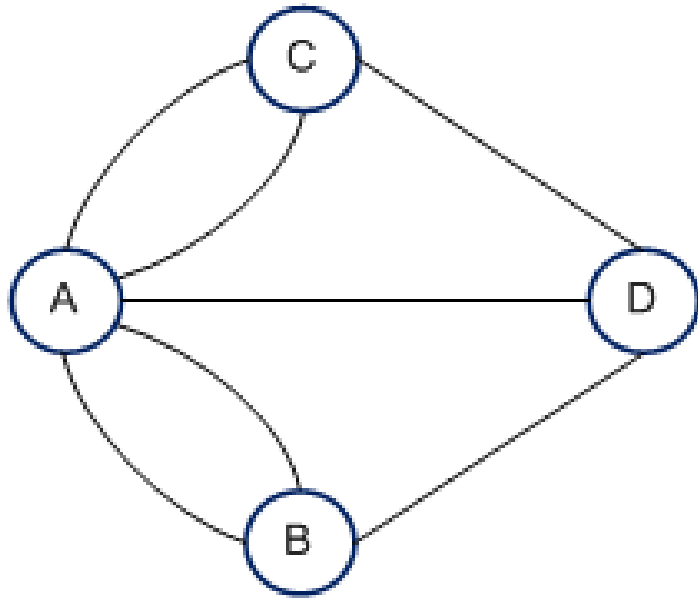
Modelare:



Problema celor 7 poduri din Königsberg

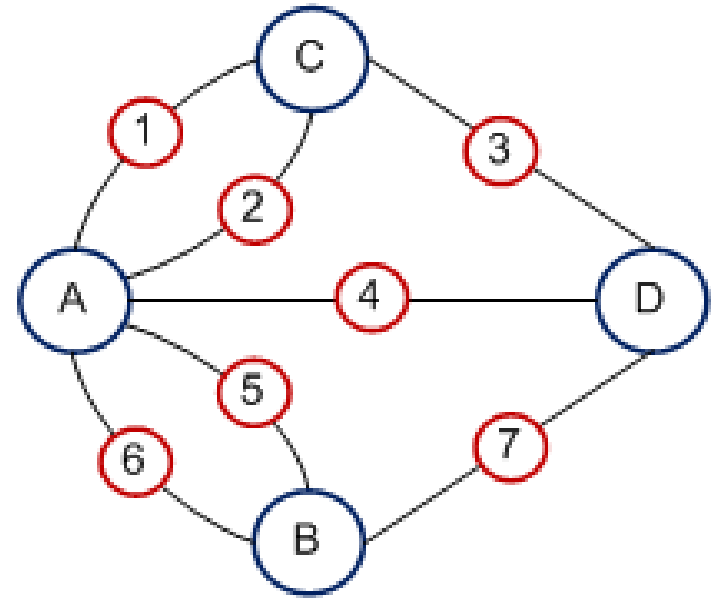
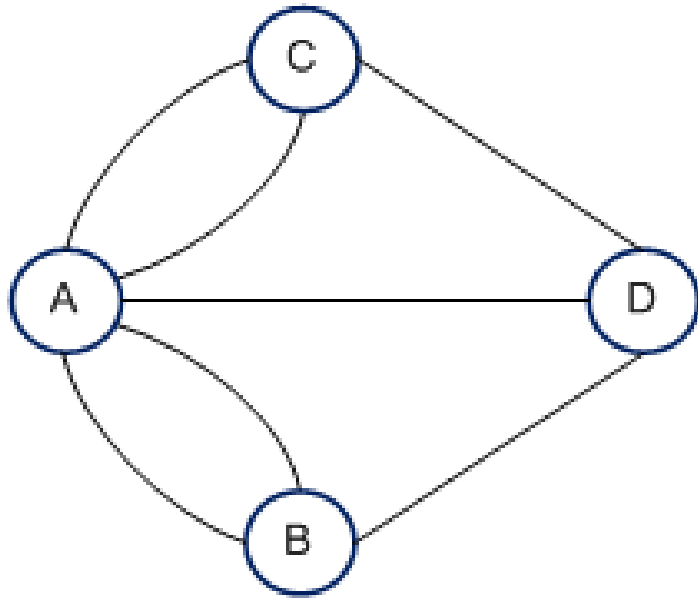


Problema celor 7 poduri din Königsberg



graf simplu

Problema celor 7 poduri din Königsberg



1736 – Leonhard Euler

Solutio problematis ad geometriam situs pertinentis

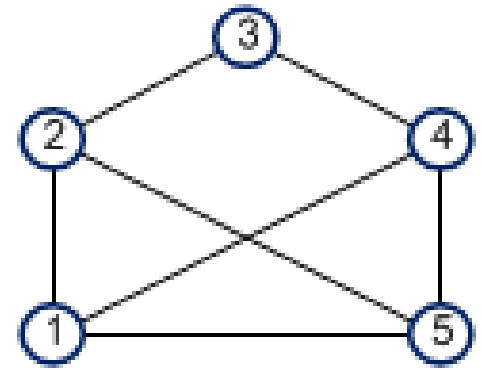
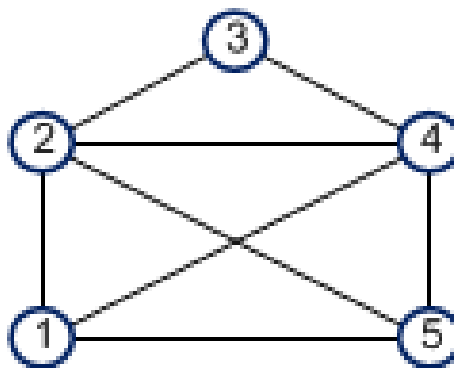
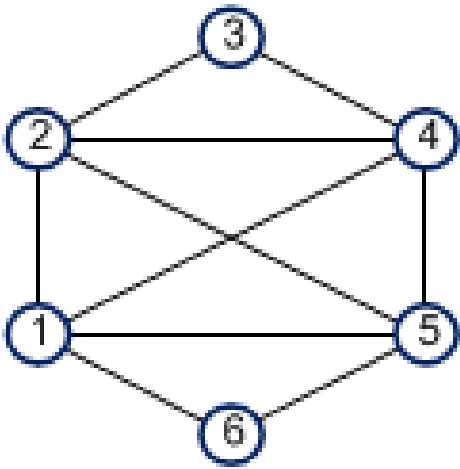
- ▶ **Ciclu eulerian** – traseu închis care trece o singură dată prin toate muchiile
- ▶ **Graf eulerian**

Problema celor 7 poduri din Königsberg

► Interpretare

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

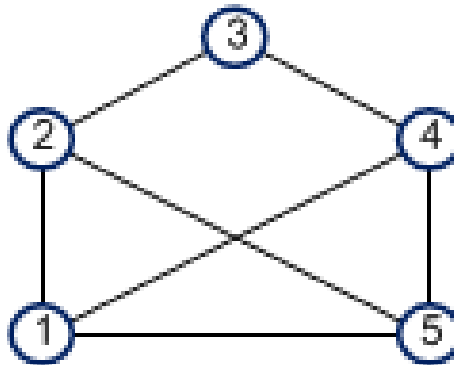
- Tăierea unui material



Problema celor 7 poduri din Königsberg

► Interpretare

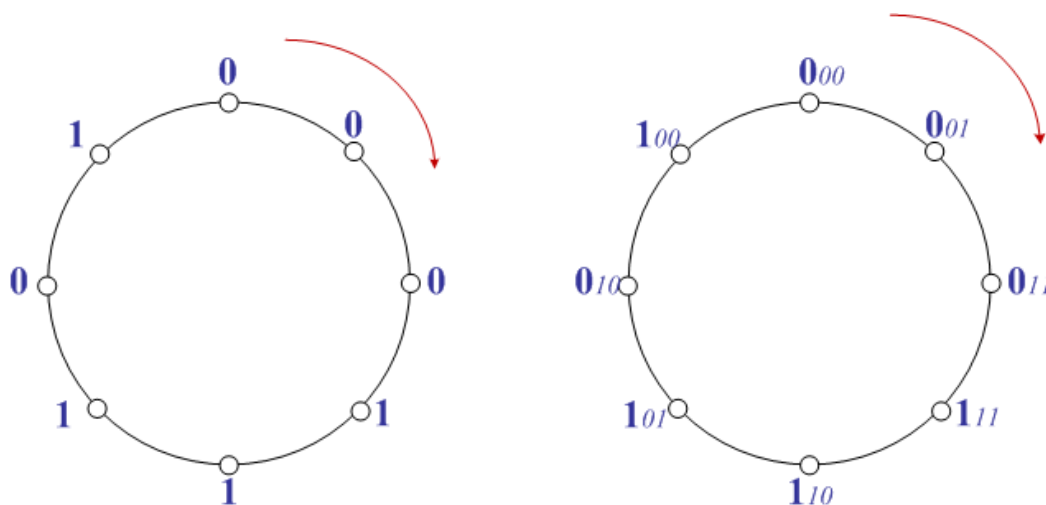
De câte ori (minim) trebuie să ridicăm creionul de pe hârtie pentru a desena diagrama?



Grafuri euleriene

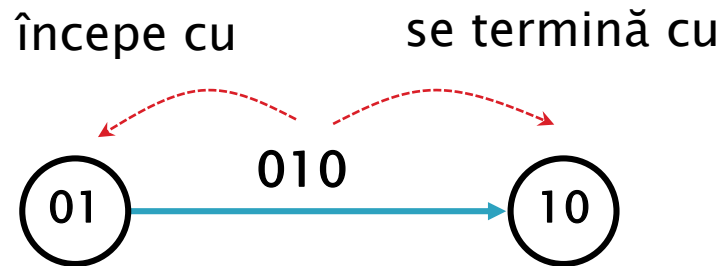
Problema lui POSTHUMUS

- ▶ $f(n)$ = 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 $f(n) \geq 2^n$. **Are loc chiar egalitate?**

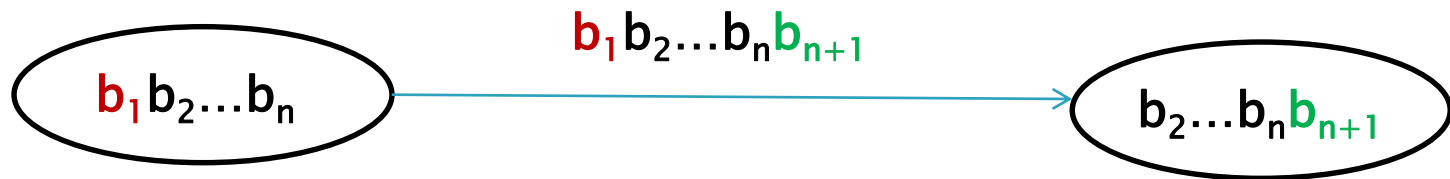


Grafuri de Bruijn

- Etichete pe arce:

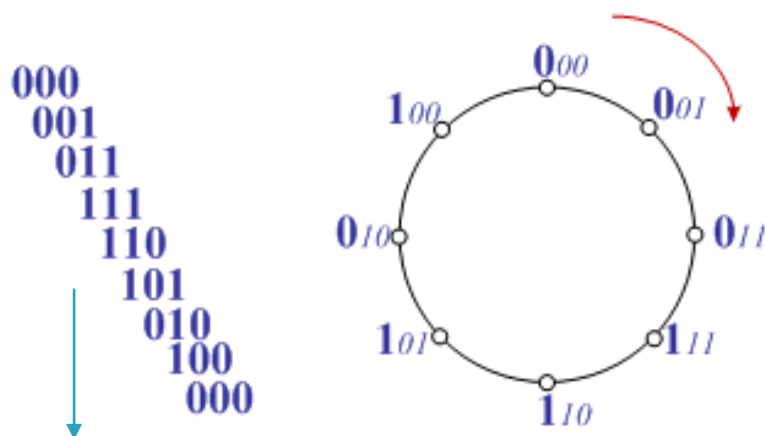
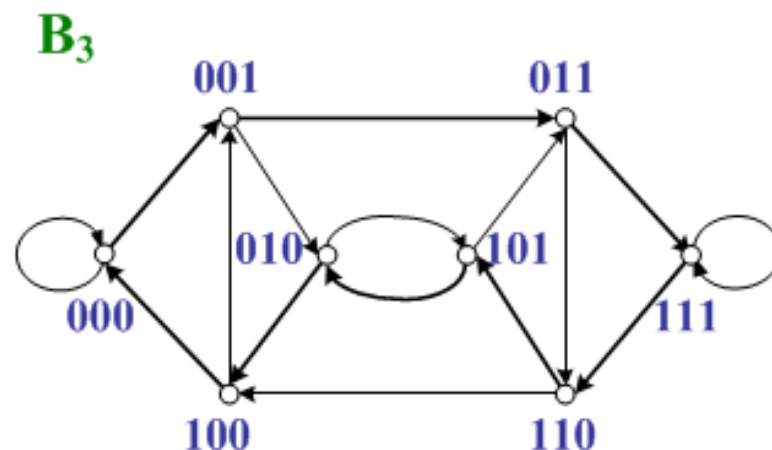
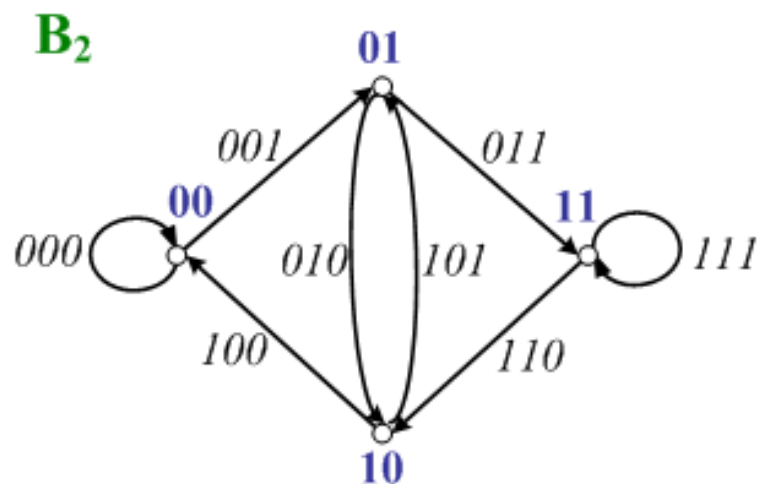


- În general:



$$b_i \in \{0,1\}$$

Grafuri de Bruijn



Soluția la problema lui POSTHUMUS pentru $n=3 \Leftrightarrow$ etichetele arcelor unui circuit eulerian în graful B_2

Grafuri euleriene

Fie G graf neorientat

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

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

- ▶ G eulerian = conține un **ciclu** eulerian

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

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

Grafuri euleriene

Observație

– Fie $P = [v_1, \dots, v_k]$

- Dacă $v_1 \neq v_k$, atunci vârfurile interne din P au gradul în P par, iar extremitățile au gradul în P impar
- Dacă $v_1 = v_k$, atunci toate vârfurile din P au gradul în P par

Grafuri euleriene

Lemă

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

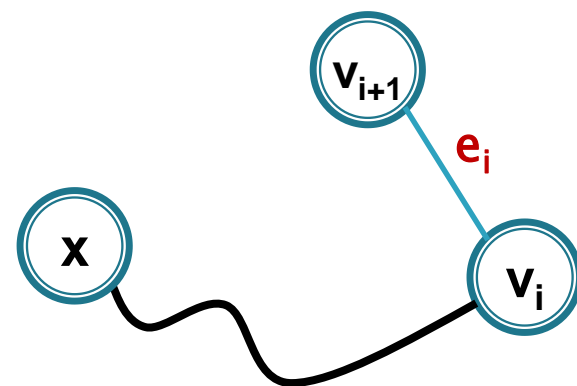
Atunci pentru orice $x \in V$ există un ciclu C în G cu $x \in V(C)$

(ciclu care conține x , nu neapărat eulerian, nici neapărat elementar)

Grafuri euleriene

Demonstrație – Algoritm de determinare a unui ciclu care conține x :

- $i = 1, v_1 = x$
- $E(C) = \emptyset$
- Repetă



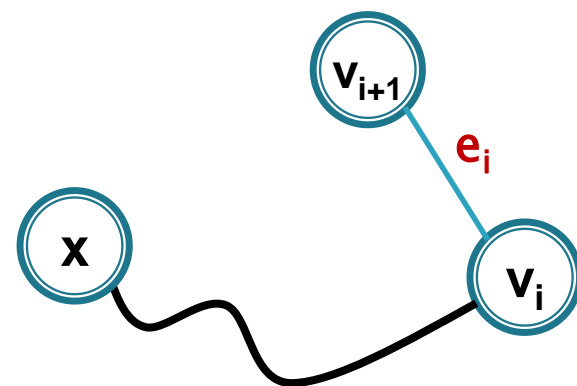
până când $v_i = x$

Grafuri euleriene

Demonstrație – Algoritm de determinare a unui ciclu care conține x :

- $i = 1, v_1 = x$
- $E(C) = \emptyset$
- Repetă
 - selectează $e_i = v_i v_{i+1} \in E(G) - E(C)$
 - $E(C) = E(C) \cup \{e_i\}$
 - $i = i + 1$

până când $v_i = x$

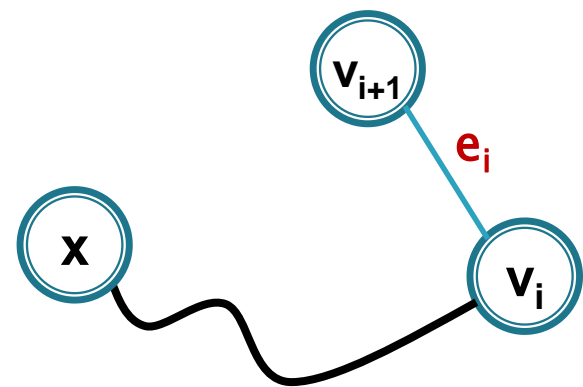


Grafuri euleriene

Demonstrație – Algoritm de determinare a unui ciclu care conține x :

- $i = 1, v_1 = x$
- $E(C) = \emptyset$
- Repetă
 - selectează $e_i = v_i v_{i+1} \in E(G) - E(C)$
 - $E(C) = E(C) \cup \{e_i\}$
 - $i = i + 1$

până când $v_i = x$



Algoritmul este corect deoarece:

Grafuri euleriene

Demonstrație – Algoritm de determinare a unui ciclu care conține x :

- $i = 1, v_1 = x$
 - $E(C) = \emptyset$
 - Repetă
 - selectează $e_i = v_i v_{i+1} \in E(G) - E(C)$ ← Dacă $v_i \neq x$, atunci $d_C(v_i)$ este impar.
 - $E(C) = E(C) \cup \{e_i\}$
 - $i = i + 1$
- până când $v_i = x$
- Din ipoteză, $d_G(v_i)$ este par
deci $d_{G-E(C)}(v_i) > 0$
 \Rightarrow muchia e_i există

Grafuri euleriene

Demonstrație – Algoritm de determinare a unui ciclu care conține x :

- $i = 1, v_1 = x$
- $E(C) = \emptyset$
- Repetă
 - selectează $e_i = v_i v_{i+1} \in E(G) - E(C)$ ← Dacă $v_i \neq x$, atunci $d_C(v_i)$ este impar.
 - $E(C) = E(C) \cup \{e_i\}$
 - $i = i + 1$

până când $v_i = x$

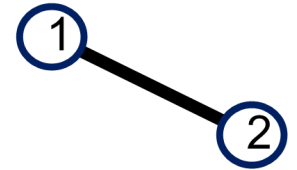
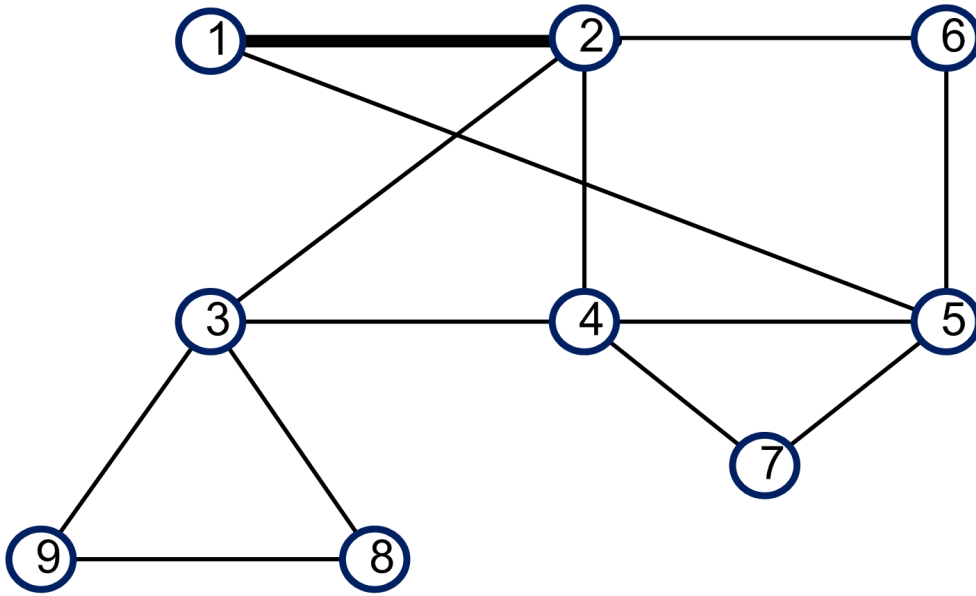
↑
 $|E(G)| < \infty$, deci algoritmul se termină (v_i ajunge egal cu x)

Din ipoteză, $d_G(v_i)$ este par
deci $d_{G-E(C)}(v_i) > 0$

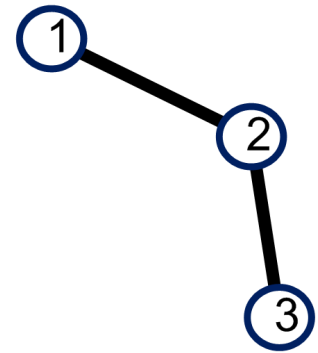
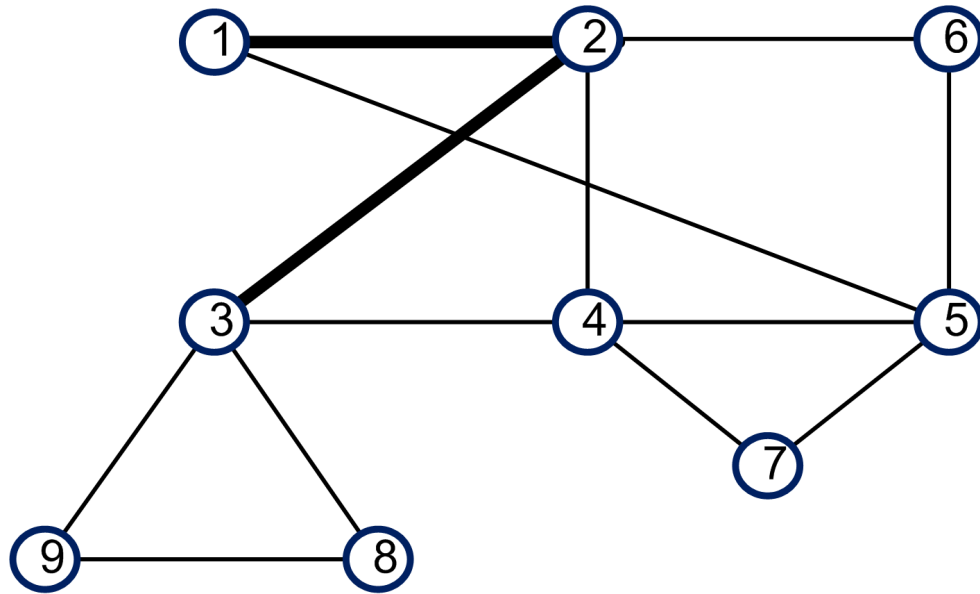
⇒ muchia e_i există

Grafuri euleriene

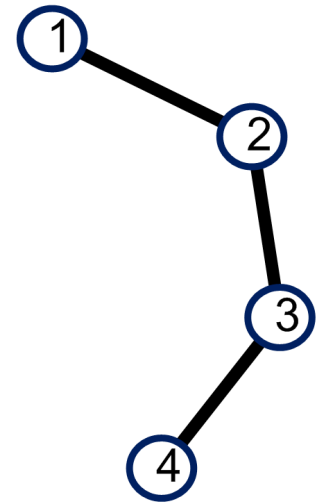
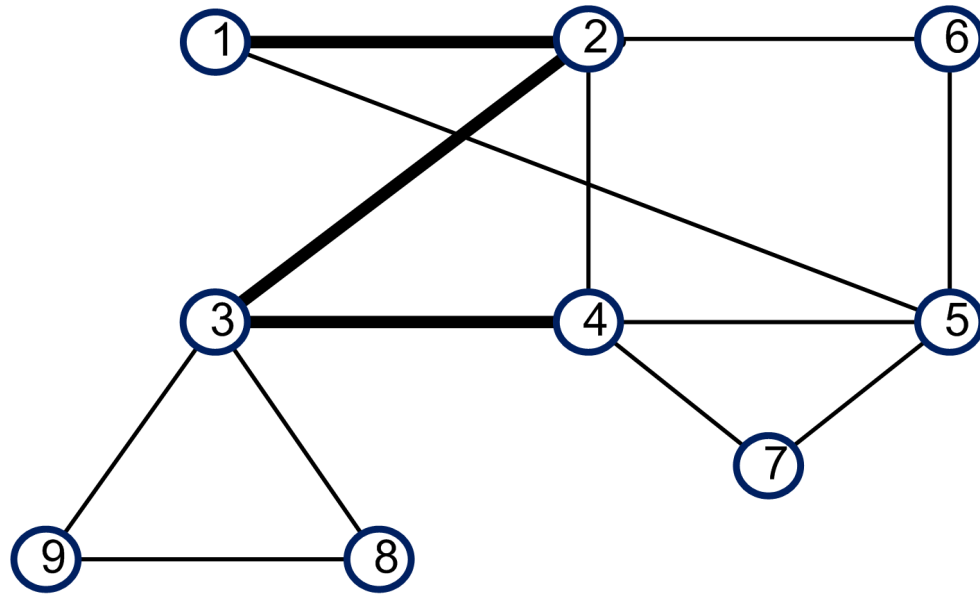
$x=1$



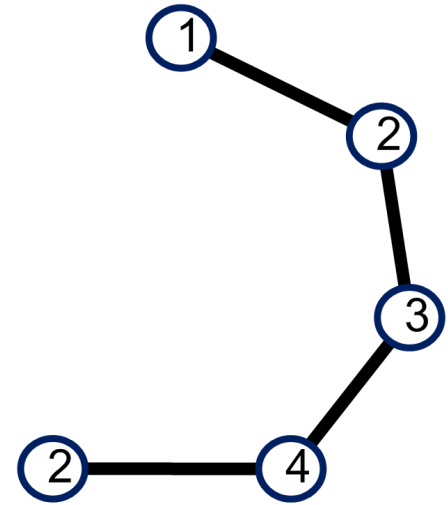
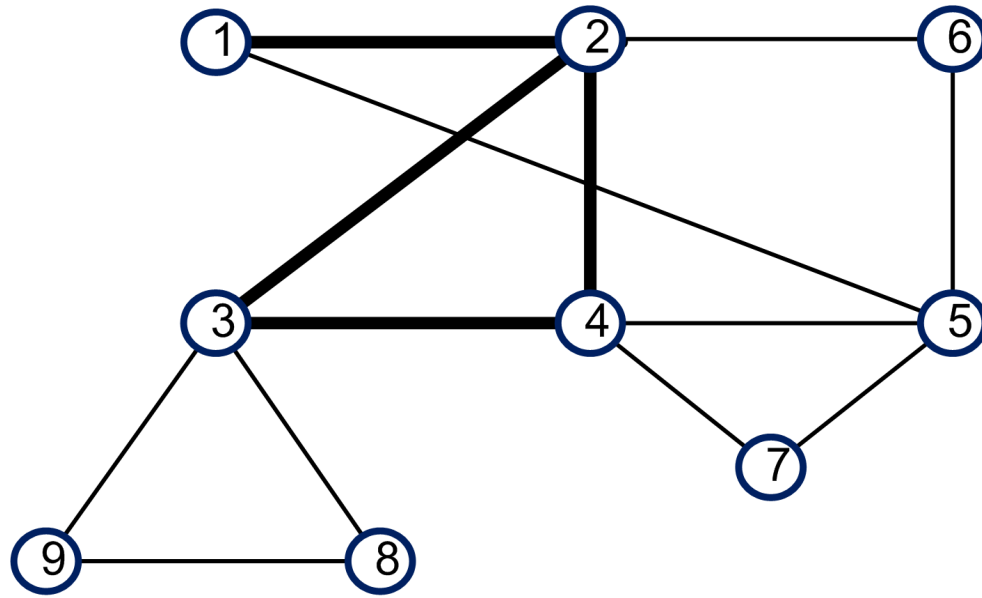
Grafuri euleriene



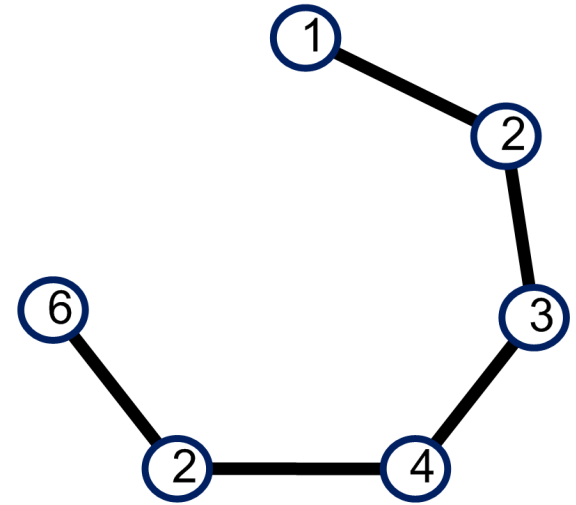
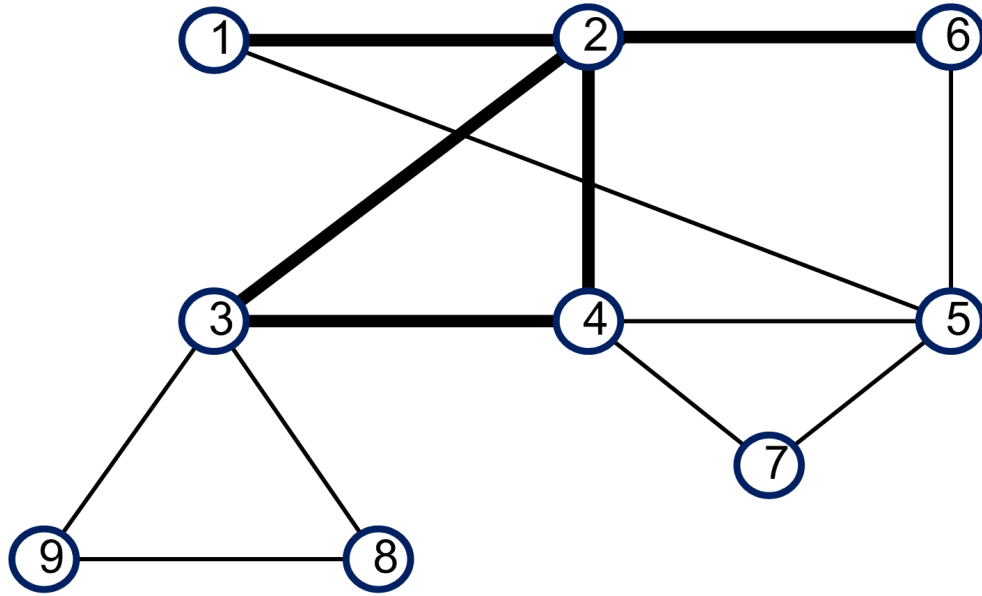
Grafuri euleriene



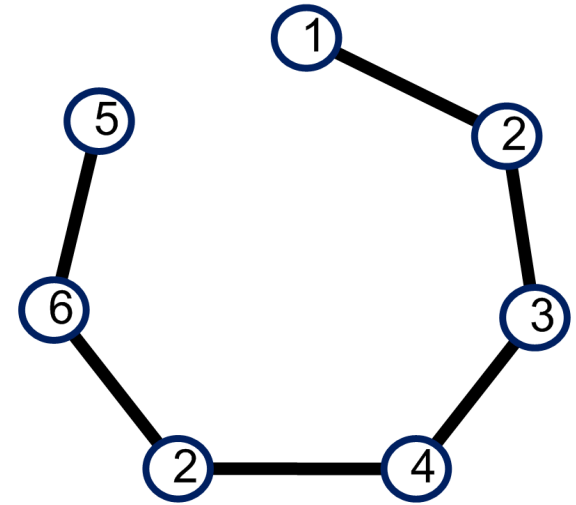
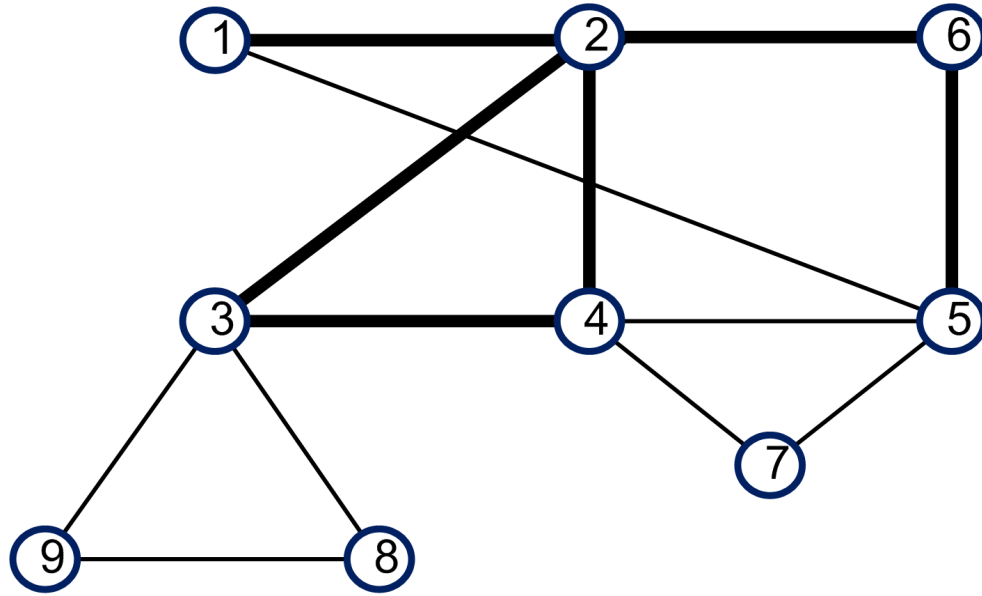
Grafuri euleriene



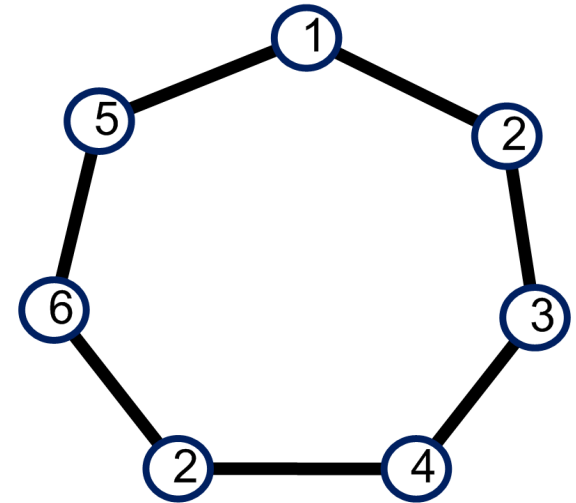
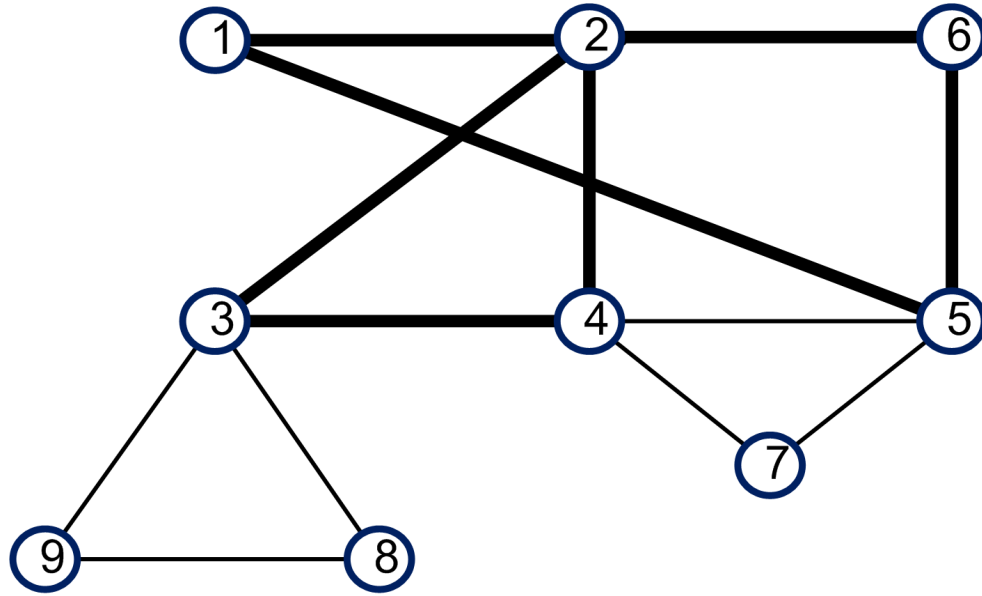
Grafuri euleriene



Grafuri euleriene



Grafuri euleriene



Grafuri euleriene

Teorema lui Euler

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

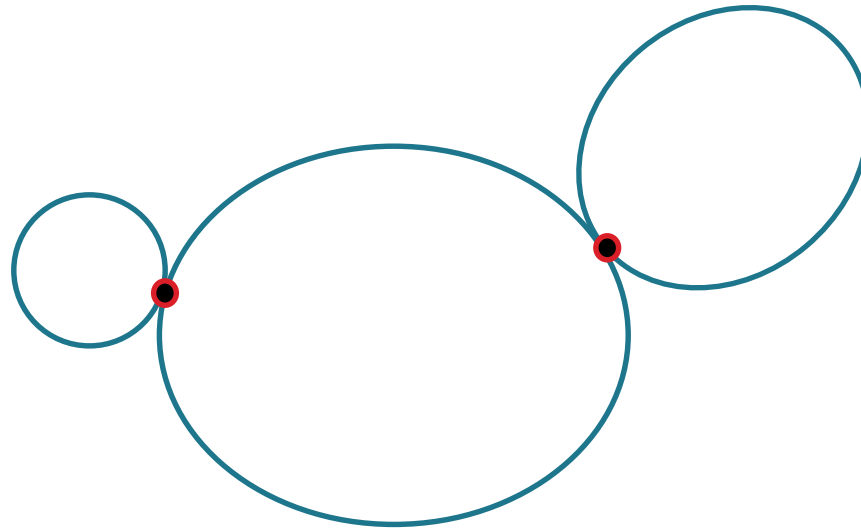
Atunci

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 $v \in V$ 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 $v \in V$ arbitrar
 - construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)
- ▶ **cât timp $|E(C)| < |E(G)|$ execută**
 - selectează $v \in V(C)$ 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 $v \in V$ arbitrar
 - construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)
- ▶ **cât timp $|E(C)| < |E(G)|$ execută**
 - selectează $v \in V(C)$ 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

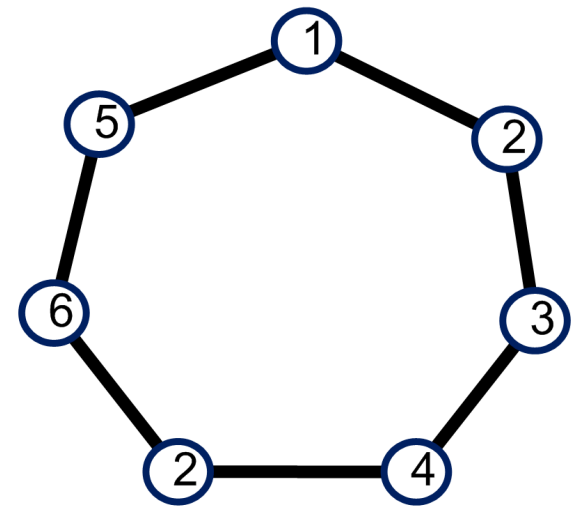
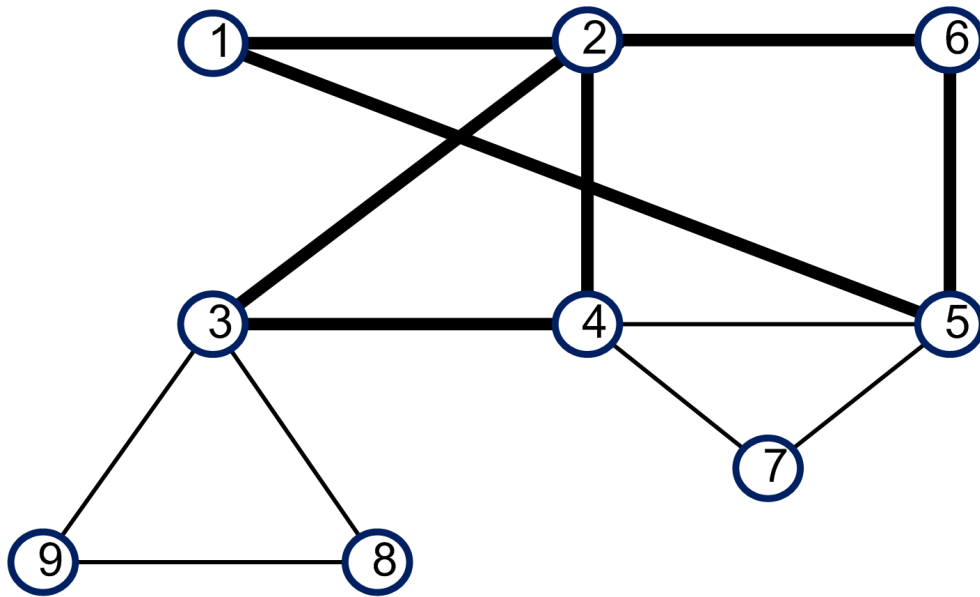
- ▶ **Pasul 0** – verificare condiții (conex+vf. izolate, grade pare)
- ▶ **Pasul 1:**
 - alege $v \in V$ arbitrar
 - construiește C un ciclu în G care începe cu v (cu algoritmul din Lema)
- ▶ **cât timp $|E(C)| < |E(G)|$ execută**
 - selectează $v \in V(C)$ cu $d_{G-E(C)}(v) > 0$ (în care sunt incidente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

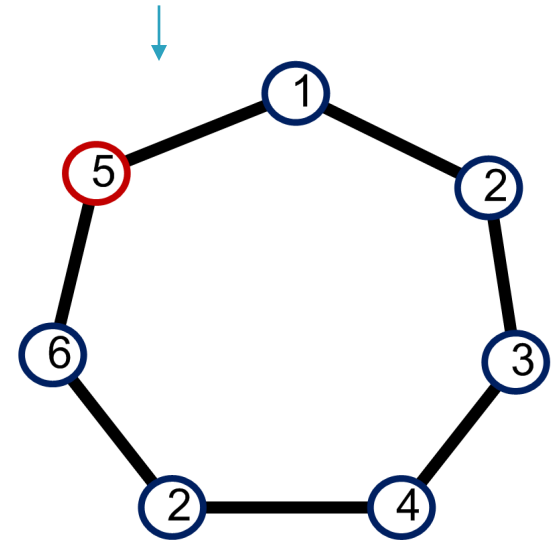
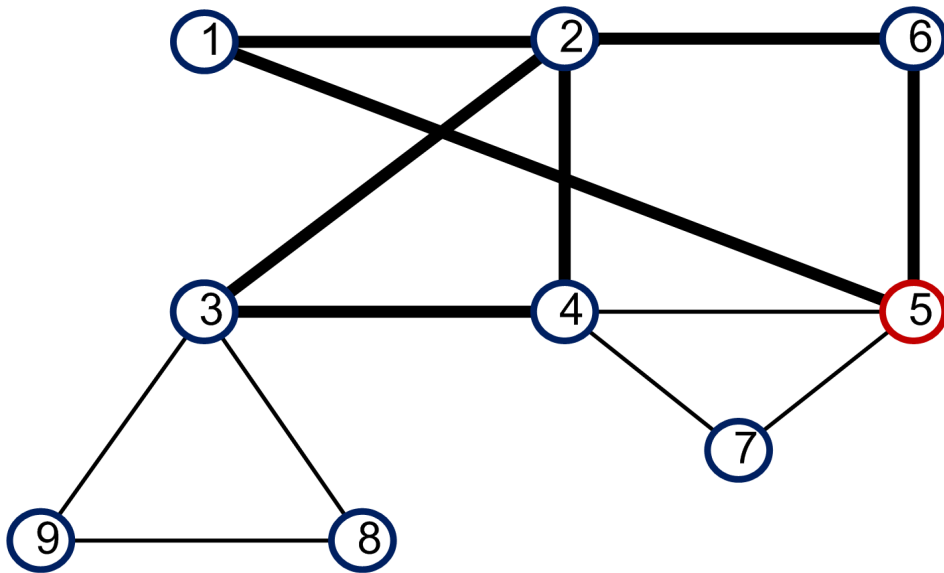
Pornim cu ciclul construit cu algoritmul din Lema 1



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

Algoritmul lui Hierholzer

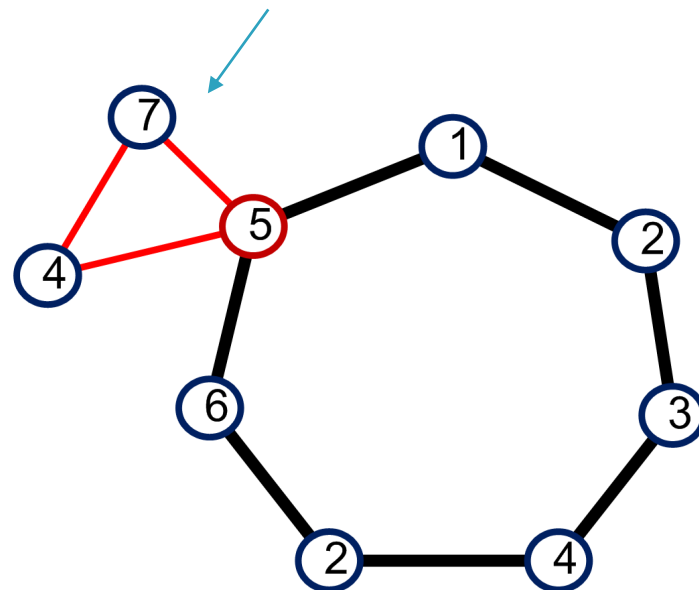
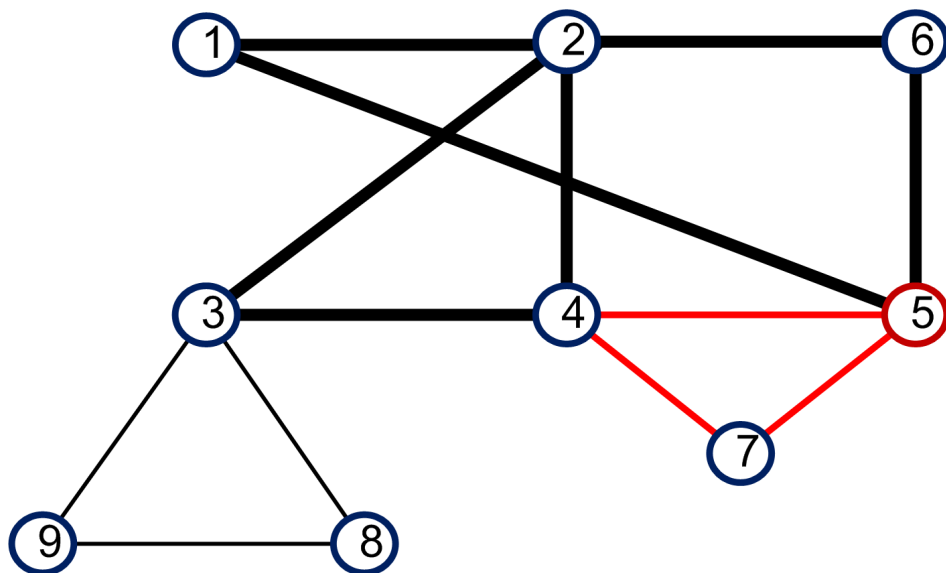
Alegem un vârf din C în care mai sunt incidente muchii, de exemplu $v = 5$



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

Algoritmul lui Hierholzer

Construim un ciclu C' cu muchiile rămase care conține $v = 5$

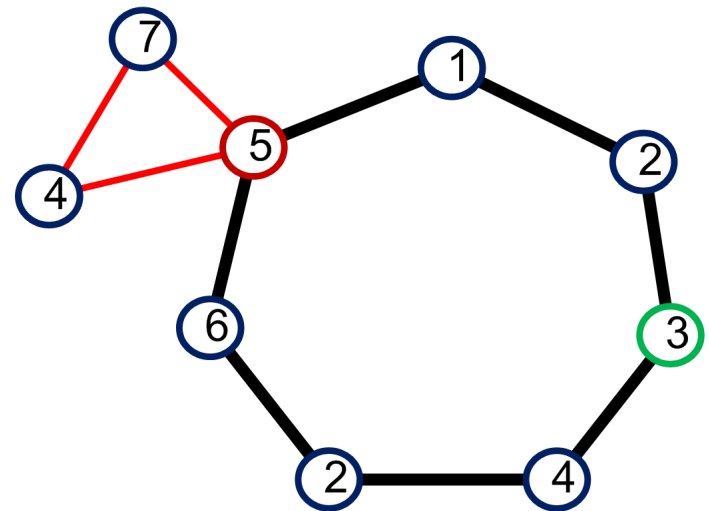
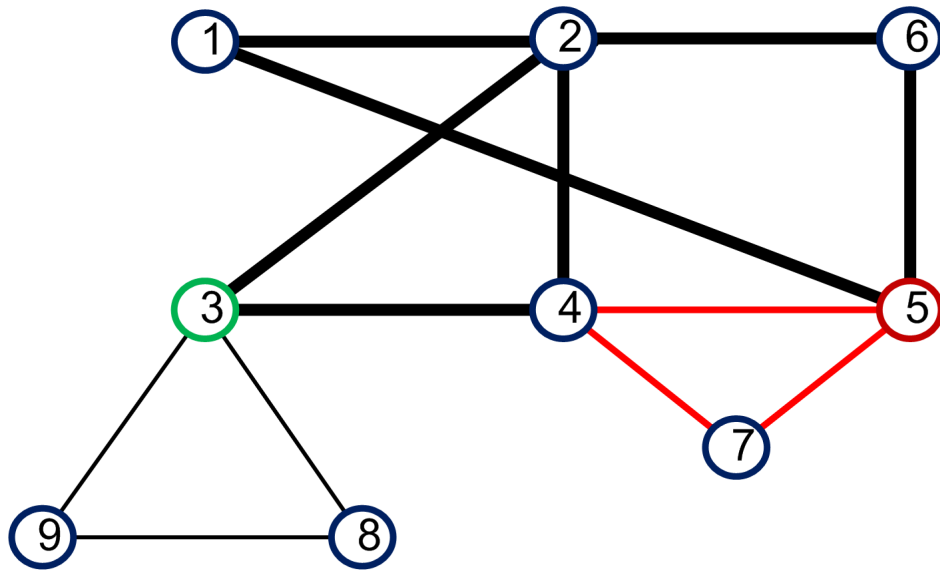


⇒ Un nou ciclu obținut prin fuziunea celor două cicluri

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

Algoritmul lui Hierholzer

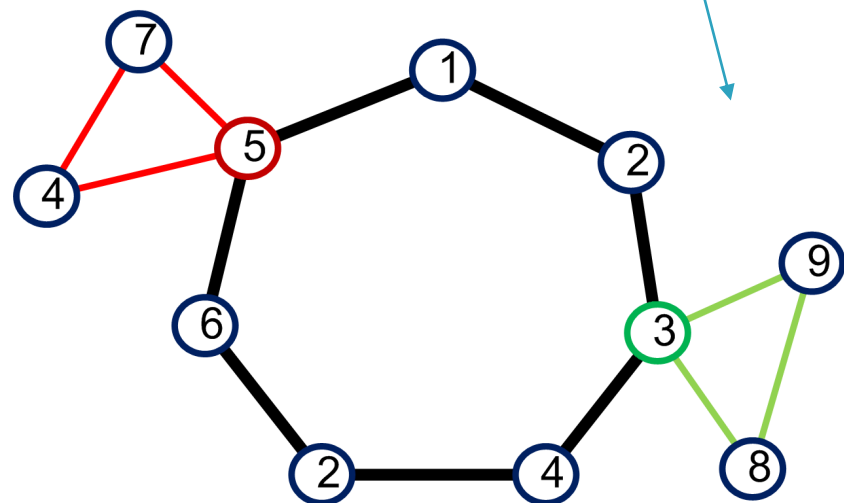
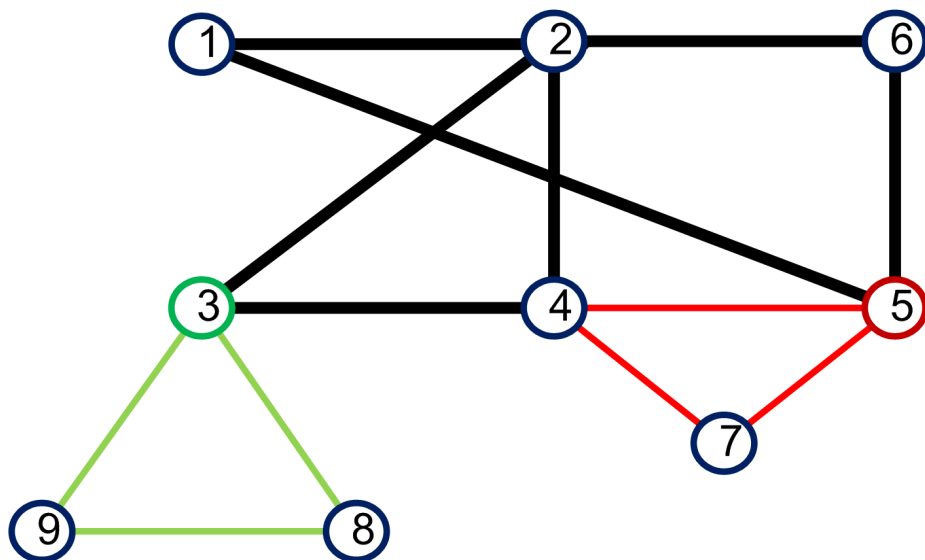
Alegem un vârf din C în care mai sunt incidente muchii, de exemplu $v = 3$



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

Algoritmul lui Hierholzer

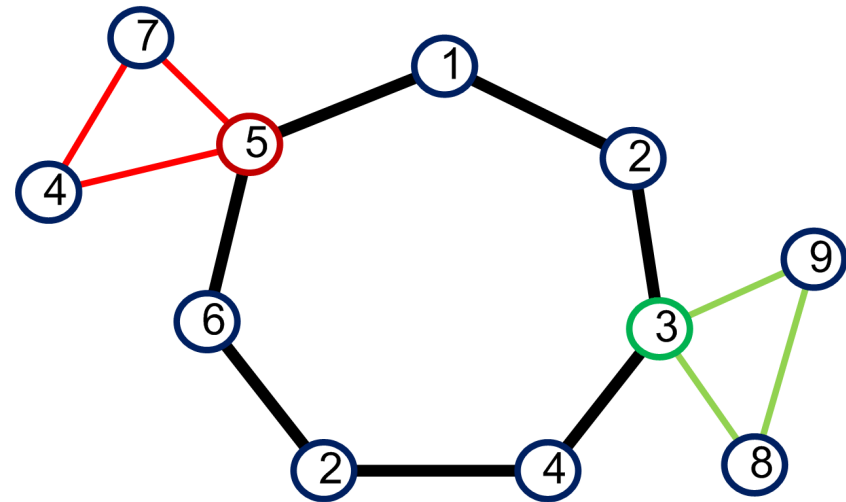
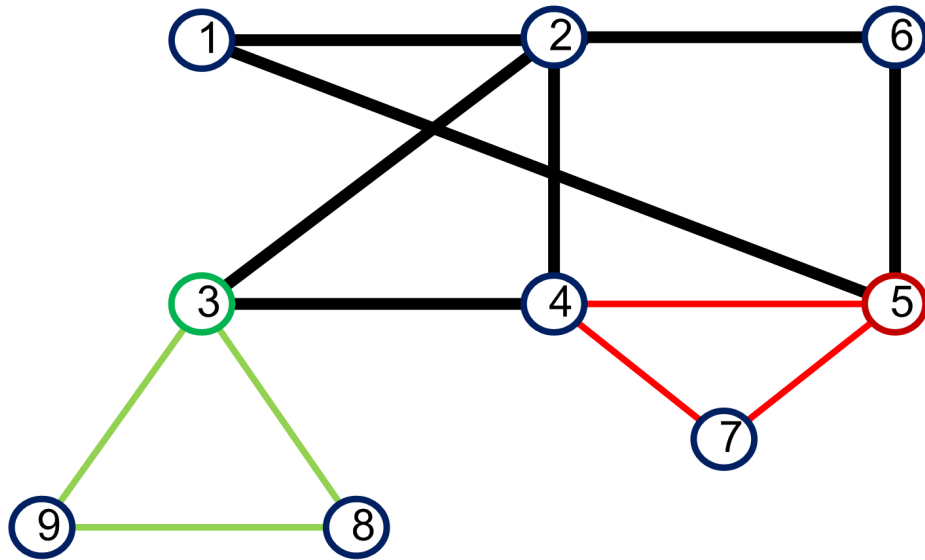
Construim un ciclu C' cu muchiile rămase care conține $v = 3$



⇒ Un nou ciclu obținut prin fuziunea celor două cicluri

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

Algoritmul lui Hierholzer



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

Ciclul conține toate muchiile

\Rightarrow este eulerian

Algoritmul lui Hierholzer

- ▶ Complexitate – $O(m)$

Algoritmul lui Hierholzer

- ▶ **Posibile implementări**

- **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 =  $\emptyset$ 
```

```
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 =  $\emptyset$ 
```

```
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 (suplimentar)

- ▶ **k-descompunere euleriană în lanțuri** a unui graf G =
o mulțime de k lanțuri simple, muchie-disjuncte

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

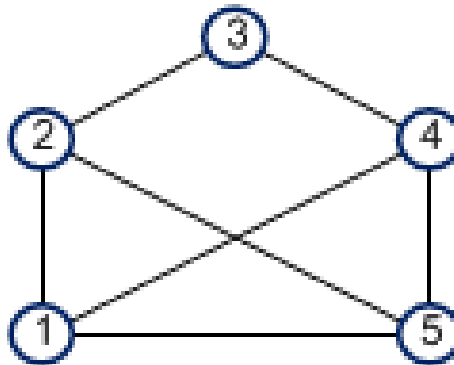
ale căror muchii induc o k -partiție a lui $E(G)$:

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

Descompuneri euleriene în lanțuri (suplimentar)

► Interpretare

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

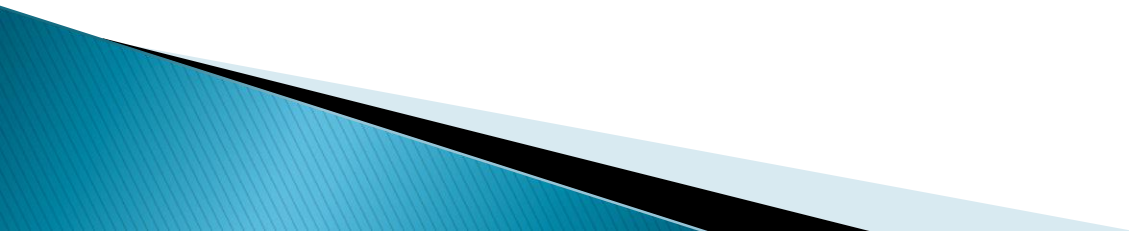


Descompuneri euleriene în lanțuri (suplimentar)

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 $E \neq \emptyset$.

Atunci

G are un drum eulerian \Leftrightarrow

$(\forall v \in V \quad d_G^-(v) = d_G^+(v))$ sau

$(\exists x \in V \text{ cu } d_G^-(x) = d_G^+(x) - 1,$

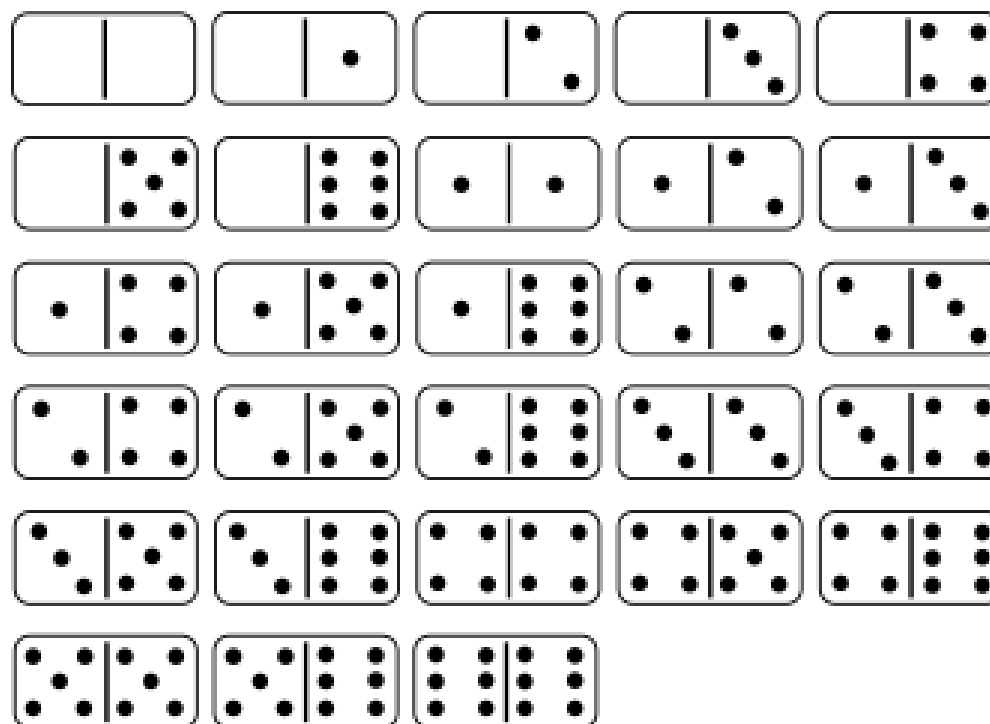
$\exists y \in V \text{ cu } d_G^-(y) = d_G^+(y) + 1,$

$\forall v \in V - \{x, y\} \quad d_G^-(v) = d_G^+(v))$

Grafuri euleriene

Problemă – joc domino

Piesă de domino – două fețe, numere $0..n$, de obicei $n=6$



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

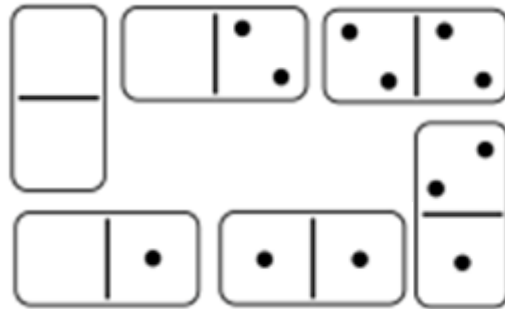
Problemă – joc domino

Se poate forma un şir de piese de domino care să conţină toate piesele + să se termine cu acelaşi număr cu care a început (un şir circular)?

Grafuri euleriene

Problemă – joc domino

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

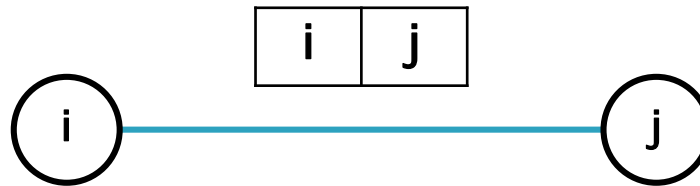


Grafuri euleriene

Problemă – joc domino

Graf asociat

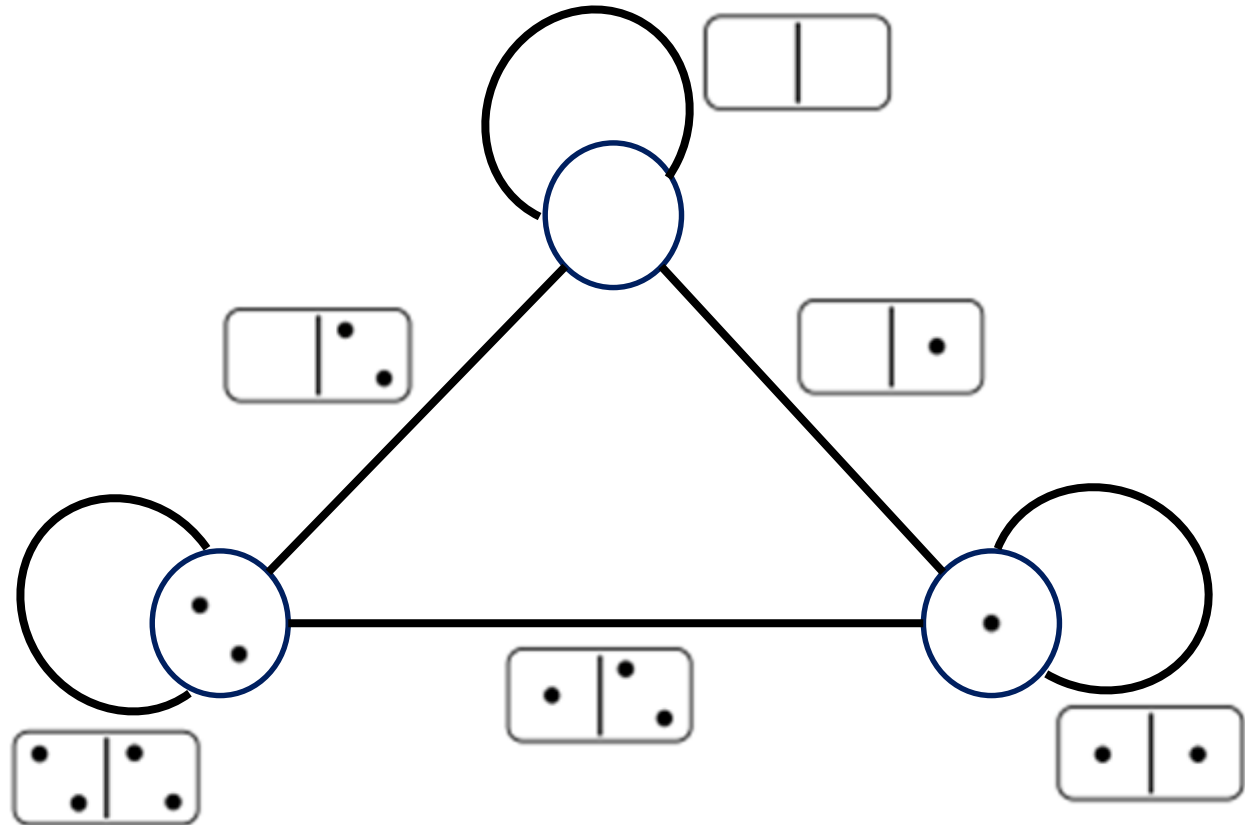
- vârfuri – numerele de pe piese
- muchii – perechi de numere (piesele)
- se pot lipi doar piese asociate muchiilor adiacente



Grafuri euleriene

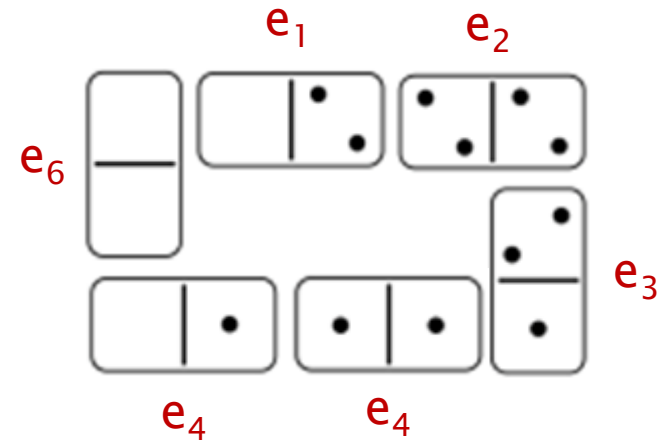
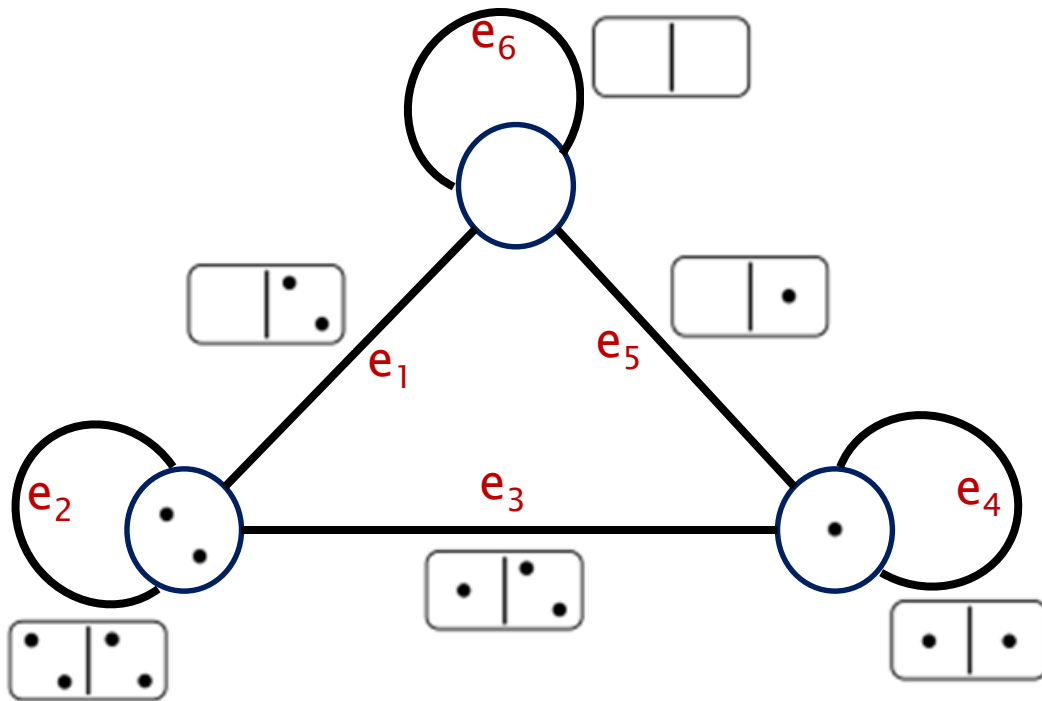
Problemă – joc domino

$n=2$



Grafuri euleriene

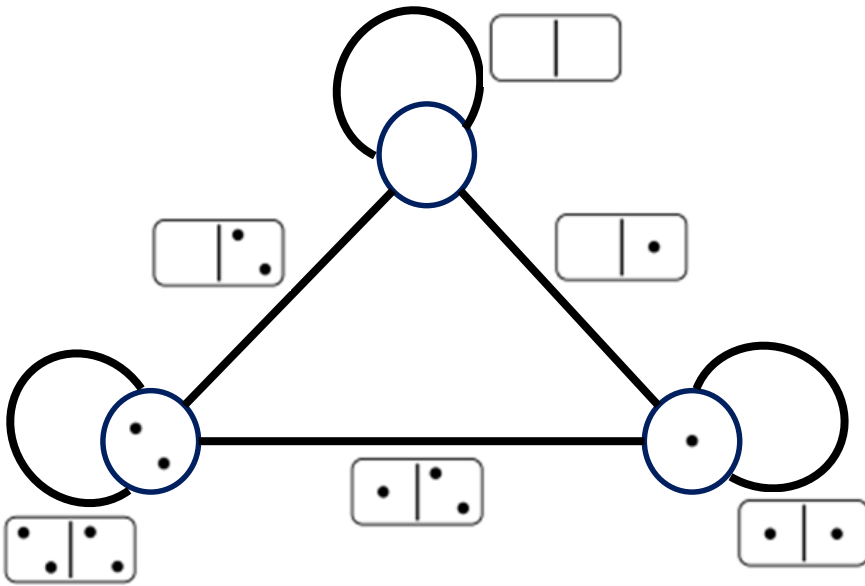
Există ciclu de piese \Leftrightarrow există ciclu eulerian în (multi)graf



Grafuri euleriene



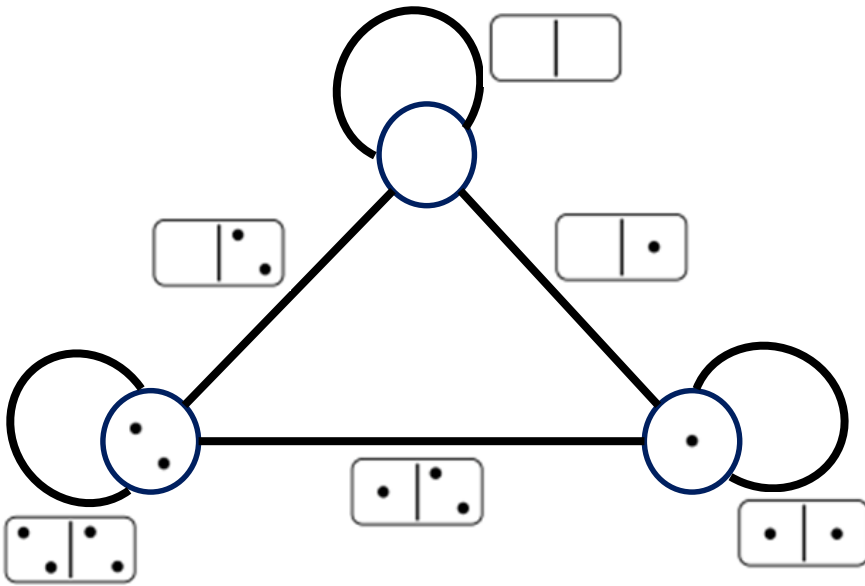
Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?



Grafuri euleriene



Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?

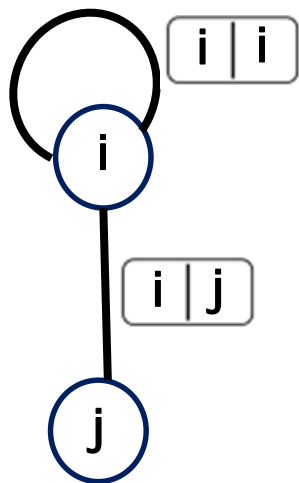


$d(i) = ?$, pentru $i = 0, \dots, n$

Grafuri euleriene



Pentru ce valori ale lui n există există ciclu eulerian în (multi)graful asociat?



$$d(i) = n+2$$

(muchii incidente în i sunt:

bucla etichetată (i,i) și

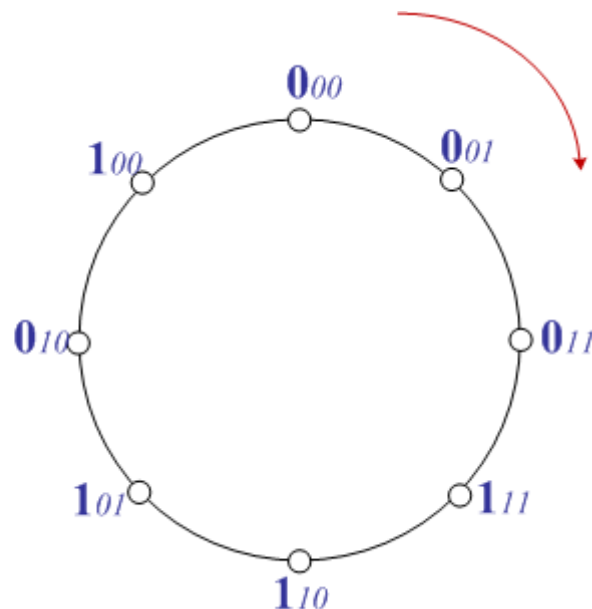
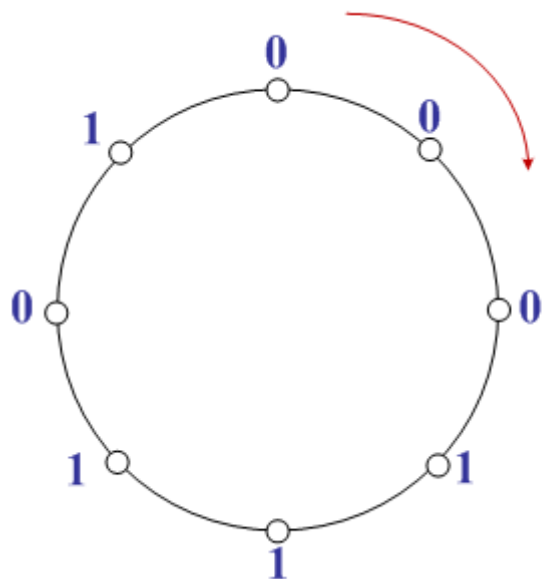
muchii etichetate $\{i,j\}$ cu $j \neq i$, $j \in \{0, \dots, n\}$)

\Rightarrow trebuie ca n să fie par

Grafuri euleriene

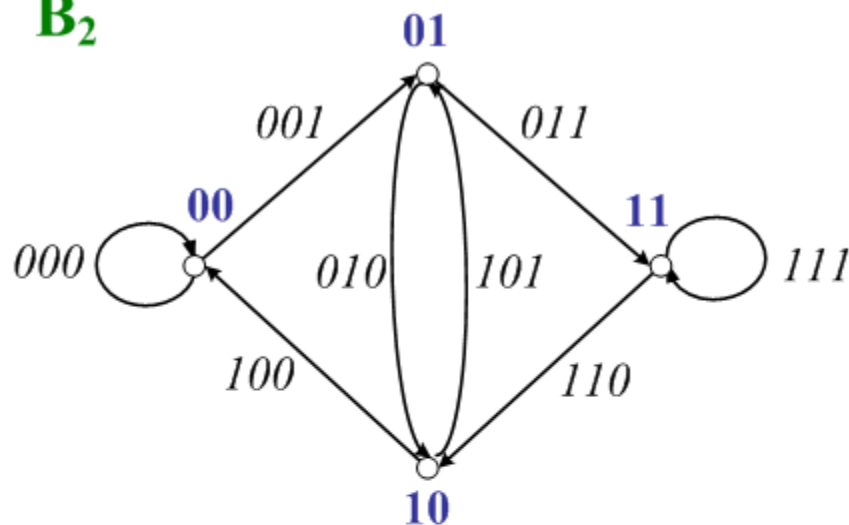
Problema lui POSTHUMUS (Suplimentar)

- ▶ $f(n)$ = 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 $f(n) \geq 2^n$. **Are loc chiar egalitate?**

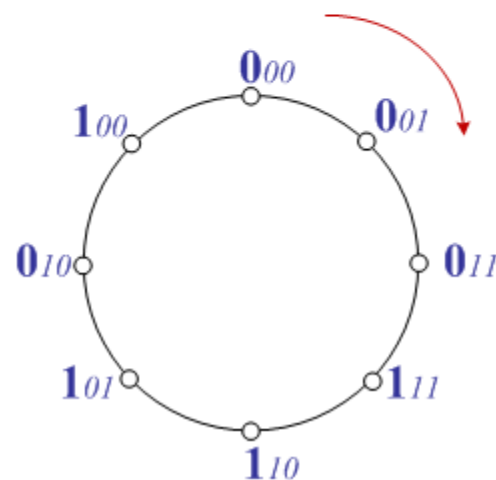


Grafuri de Bruijn

B_2



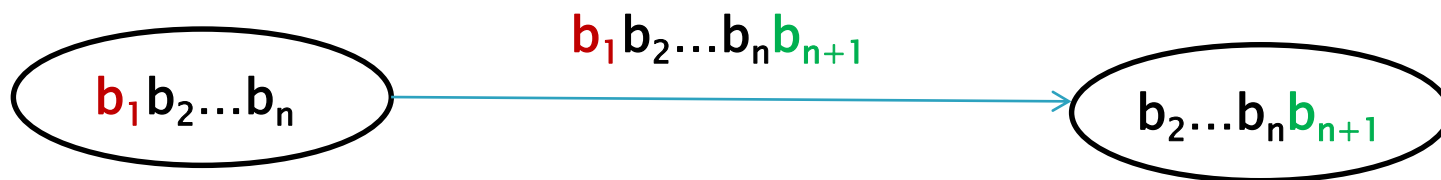
000
001
011
111
110
101
010
100
000



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

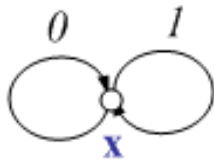
Grafuri de Bruijn

- ▶ 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 $\{0,1\}^{n+1}$ ($\{0,1,\dots,p\}^{n+1}$)
 $b_1b_2\dots b_nb_{n+1}$ etichetează arcul de la
 $b_1b_2\dots b_n$ la $b_2\dots b_nb_{n+1}$

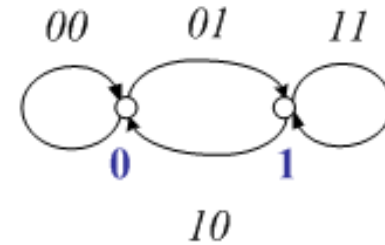


Grafuri de Bruijn

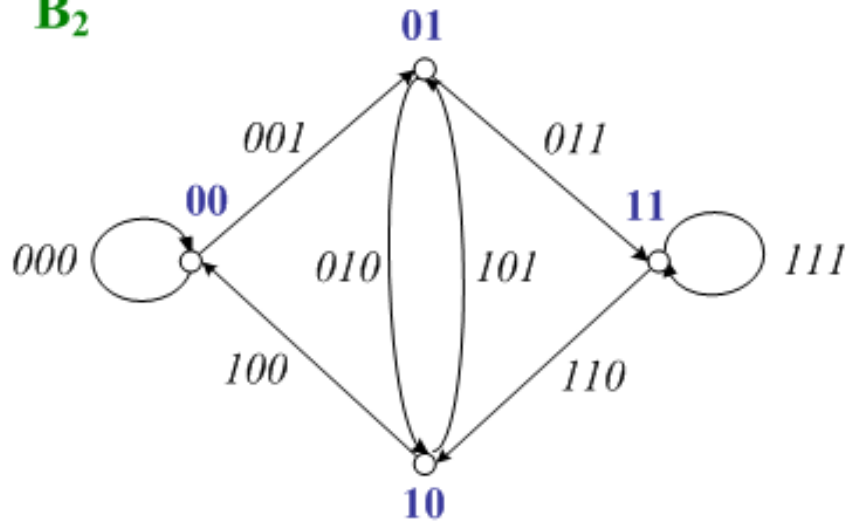
B_0



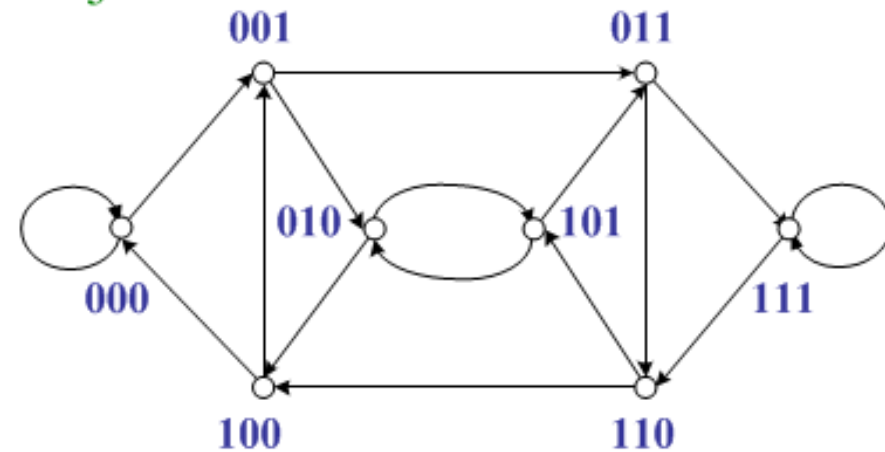
B_1



B_2



B_3

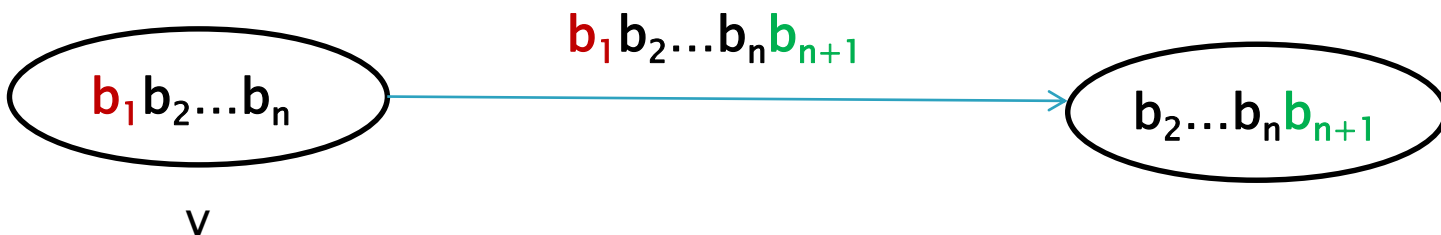


Grafuri de Bruijn

- ▶ B_n este eulerian?

$$d^+(v) = ?$$

$$d^-(v) = ?$$



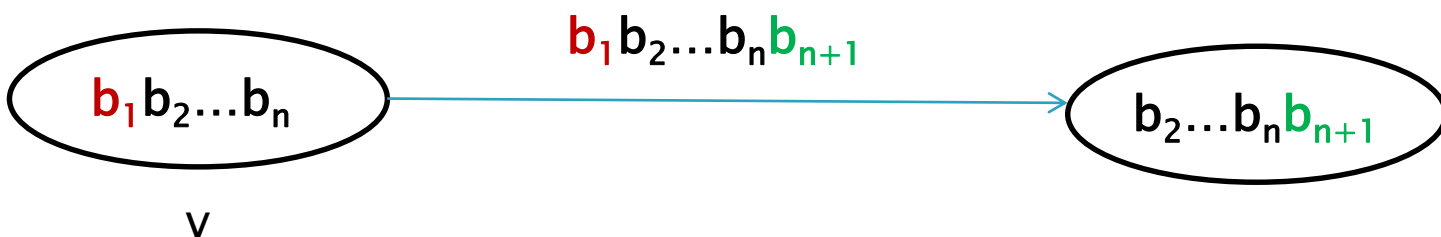
orice b_{n+1} din alfabet

Grafuri de Bruijn

- ▶ B_n este eulerian

$$d^+(v) = |\{0,1\}| = 2 \quad (\text{mai general } = |\{0,1,\dots, p\}|)$$

$$d^-(v) = d^+(v)$$



orice b_{n+1} din alfabet

Grafuri de Bruijn

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

$$f(n) = 2^n$$

- ▶ Observație

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

Grafuri de Bruijn

- ▶ Aplicație – genetică (*Genome Assembly*)