Algoritmica Grafurilor

Secvențe de Grade

Secvențe de Grade

- **Dată o secvenţă de numere s, se poate construi un graf neorientat având secvenţa gradelor s?**
- **Dar un multigraf neorientat?**
- **Dar un arbore?**

  - **Condiţii necesare**
  - **Condiţii suficiente**

Construcția de grafuri neorientate cu secvența gradelor dată.

Algoritmul Havel-Hakimi

Construcția de grafuri neorientate cu secvența gradelor dată.

Problemă:

Fie $s_0 = \{d_1, \ldots, d_n\}$ o secvență de numere naturale.

Să se construiască, dacă se poate, un graf neorientat G cu $s(G) = s_0$.

Construcția de grafuri neorientate cu secvența gradelor dată.

Problemă:

Fie $s_0 = \{d_1, \ldots, d_n\}$ o secvență de numere naturale.

Să se construiască, dacă se poate, un graf neorientat G cu

$s(G) = s_0$.

Condiții <u>necesare</u> pentru existența lui G:

- **$d_1 + \ldots + d_n$ – număr par $0 \leq d_i \leq n - 1$**

Construcția de grafuri neorientate cu secvența gradelor dată.

Problemă:

Fie $s_0 = \{d_1, \ldots, d_n\}$ o secvență de numere naturale.

Să se construiască, dacă se poate, un graf neorientat G cu

$s(G) = s_0$.

Condiții <u>necesare</u> pentru existența lui G:

- **$d_1 + \ldots + d_n$ – număr par $0 \leq d_i \leq n - 1$**

Pentru $s_0 = \{3, 3, 1, 1\}$ - nu există G

condițiile nu sunt și suficiente

Construcția de grafuri neorientate cu secvența gradelor dată. Algoritm.

Idee algoritm de construcție a unui graf G cu $s(G) = s_0$

- **începem construcția de la vârful cu gradul cel mai mare**

Construcția de grafuri neorientate cu secvența gradelor dată. Algoritm.

Idee algoritm de construcție a unui graf G cu $s(G) = s_0$

- **începem construcția de la vârful cu gradul cel mai mare**
- **îi alegem ca vecini vârfurile cu gradele cele mai mari**

Construcția de grafuri neorientate cu secvența gradelor dată. Algoritm.

Idee algoritm de construcție a unui graf G cu s(G) =s_0

- **începem construcția de la vârful cu gradul cel mai mare**
- **îi alegem ca vecini vârfurile cu gradele cele mai mari**
- **actualizăm secvența s_0 și reluăm până când**

Construcția de grafuri neorientate cu secvența gradelor dată. Algoritm.

Idee algoritm de construcție a unui graf G cu s(G) =s_0

- **începem construcția de la vârful cu gradul cel mai mare**
- **îi alegem ca vecini vârfurile cu gradele cele mai mari**
- **actualizăm secvența s_0 și reluăm până când**
  - **secvența conține doar 0 ⇒ G**

    sau

  - **secvența conține numere negative ⇒**

G nu se poate construi prin acest procedeu

Exemplu Algoritm

Fie S_0 = { 3, 4, 2, 1, 3, 4, 1, 2}

etichetele nodurilor x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8

x_7

Exemplu Algoritm

Fie S_0 = { 3, 4, 2, 1, 3, 4, 1, 2}

etichetele nodurilor x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8

Pasul 1 - construim muchii pentru vârful de gradul maxim = x_2

- alegem ca vecini următoarele vârfuri cu cele mai mari grade

Exemplu Algoritm

Fie S_0 = { 3, 4, 2, 1, 3, 4, 1, 2}

etichetele nodurilor x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8

Pasul 1 - construim muchii pentru vârful de gradul maxim = x_2

- alegem ca vecini următoarele vârfuri cu cele mai mari grade

$\Rightarrow$ **ar fi utilă sortarea descrescătoare a elementelor lui s_0**

S_0 = { 4, 4, 3, 3, 2, 2, 1, 1}

etichete noduri x_2 x_6 x_1 x_5 x_3 x_8 x_4 x_7

Exemplu Algoritm

Exemplu algoritm Havel–Hakimi

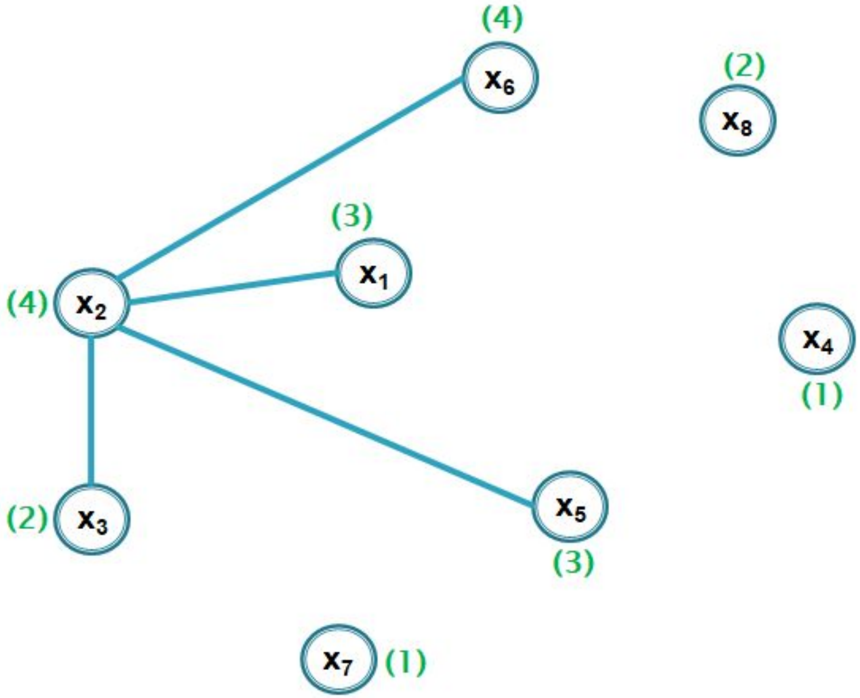
Pasul 1.

$s_0 = \{$	4,	4,	3,	3,	2,	2,	1,	$1\}$
etichete vârfuri	x_2	x_6	x_1	x_5	x_3	x_8	x_4	x_7

- Muchii construite: $x_2x_6, x_2x_1, x_2x_5, x_2x_3$

Exemplu Algoritm

Exemplu algoritm Havel–Hakimi



Exemplu Algoritm

Exemplu algoritm Havel–Hakimi

Pasul 1.

$s_0 = \{\ 4,\ 4,\ 3,\ 3,\ 2,\ 2,\ 1,\ 1\}$

etichete vârfuri $x_2\ \ x_6\ \ x_1\ \ x_5\ \ x_3\ \ x_8\ \ x_4\ \ x_7$

- Muchii construite: $x_2x_6, x_2x_1, x_2x_5, x_2x_3$
- Secvența rămasă:

$s'_0 = \{\ 3,\ 2,\ 2,\ 1,\ 2,\ 1,\ 1\}$

etichete vârfuri $x_6\ \ x_1\ \ x_5\ \ x_3\ \ x_8\ \ x_4\ \ x_7$

Exemplu Algoritm

Exemplu algoritm Havel–Hakimi

Pasul 1.

$s_0 = \{\ 4,\ 4,\ 3,\ 3,\ 2,\ 2,\ 1,\ 1\}$

etichete vârfuri $x_2\ x_6\ x_1\ x_5\ x_3\ x_8\ x_4\ x_7$

- Muchii construite: $x_2x_6, x_2x_1, x_2x_5, x_2x_3$
- Secvența rămasă:

$s'_0 = \{\ 3,\ 2,\ 2,\ 1,\ 2,\ 1,\ 1\}$

etichete vârfuri $x_6\ x_1\ x_5\ x_3\ x_8\ x_4\ x_7$

Secvența rămasă ordonată descrescător:

$s'_0 = \{\ 3,\ 2,\ 2,\ 2,\ 1,\ 1,\ 1\}$

etichete vârfuri $x_6\ x_1\ x_5\ x_8\ x_3\ x_4\ x_7$

Exemplu Algoritm

Exemplu algoritm Havel–Hakimi

Pasul 2.

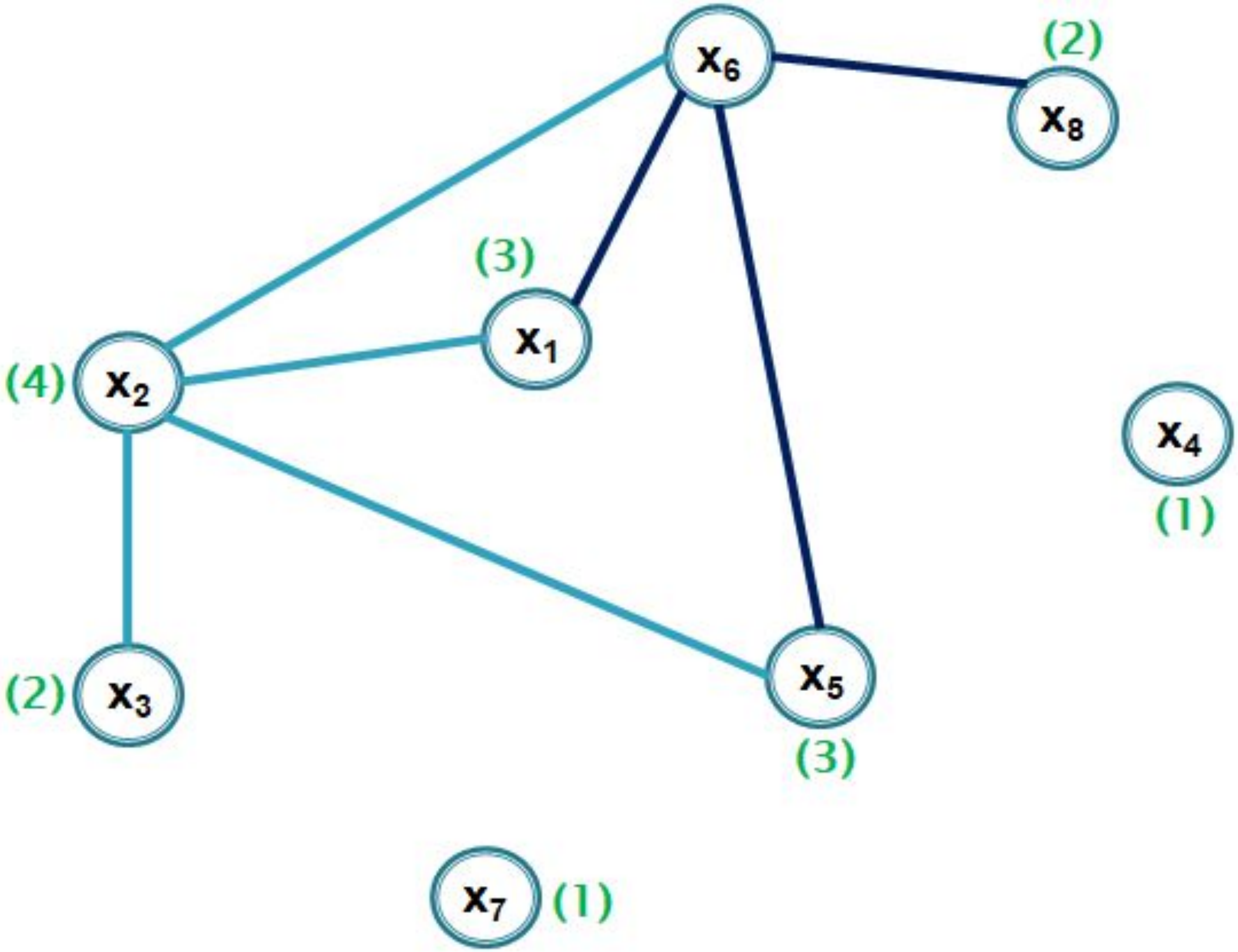
$s'_0 = \{$	3,	2,	2,	2,	1,	1,	1}
etichete vârfuri	x_6	x_1	x_5	x_8	x_3	x_4	x_7

- Muchii construite: x_6x_1, x_6x_5, x_6x_8
- Secvența rămasă:

$s''_0 = \{$	1,	1,	1,	1,	1,	1}
etichete vârfuri	x_1	x_5	x_8	x_3	x_4	x_7

(este ordonată descrescător)

Exemplu Algoritm



Exemplu Algoritm

Pasul 3.

$s''_0 = \{1, 1, 1, 1, 1, 1\}$

etichete vârfuri x_1 x_5 x_8 x_3 x_4 x_7

- Muchii construite: x_1x_5
- Secvența rămasă:

$s'''_0 = \{0, 1, 1, 1, 1\}$

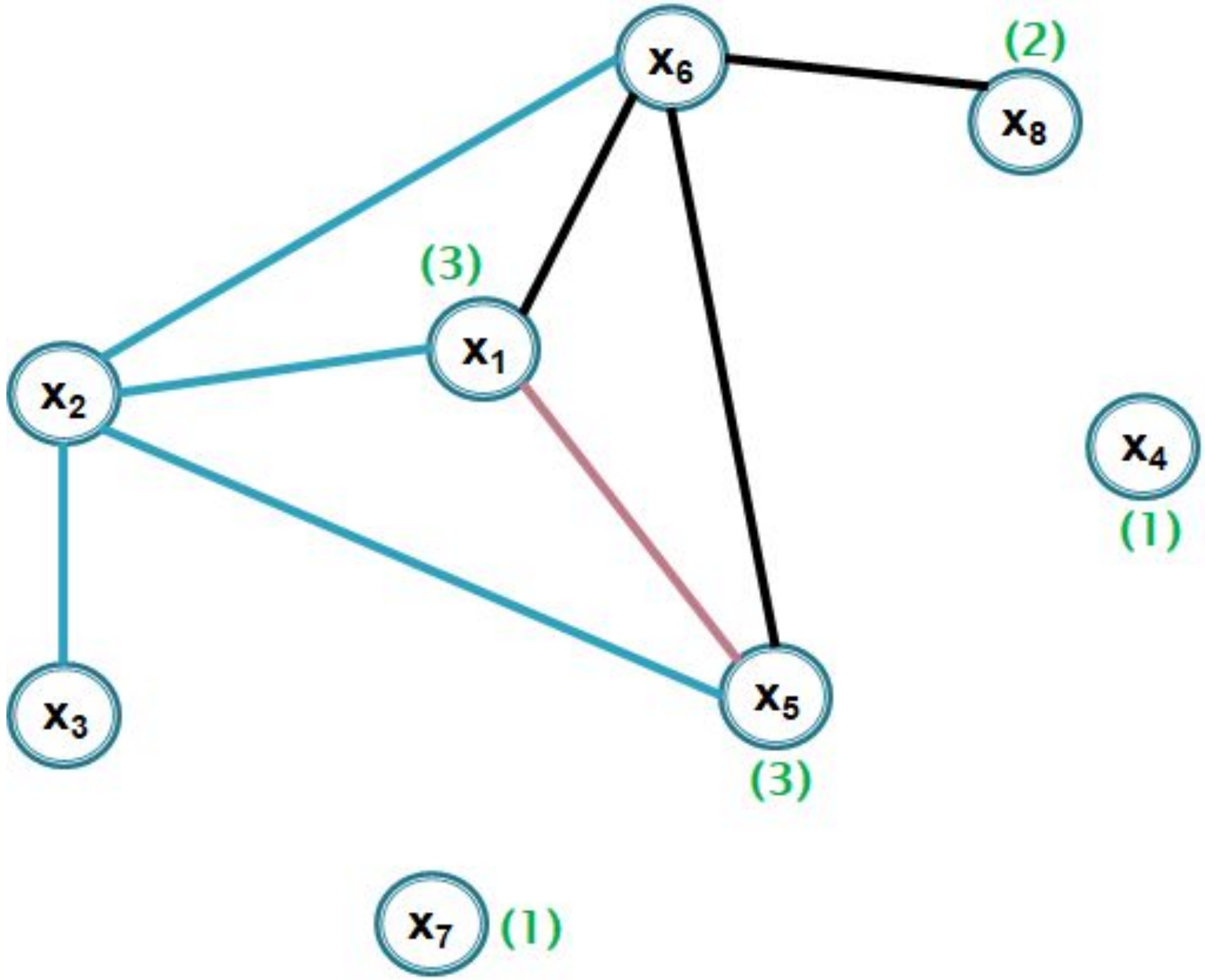
etichete vârfuri x_5 x_8 x_3 x_4 x_7

Secvența rămasă ordonată descrescător:

$s'''_0 = \{1, 1, 1, 1, 0\}$

etichete vârfuri x_7 x_3 x_4 x_8 x_5

Exemplu Algoritm



Exemplu Algoritm

Pasul 4.

$s'''_0 = \{$	1,	1,	1,	1,	0}
etichete vârfuri	x_7	x_3	x_4	x_8	x_5

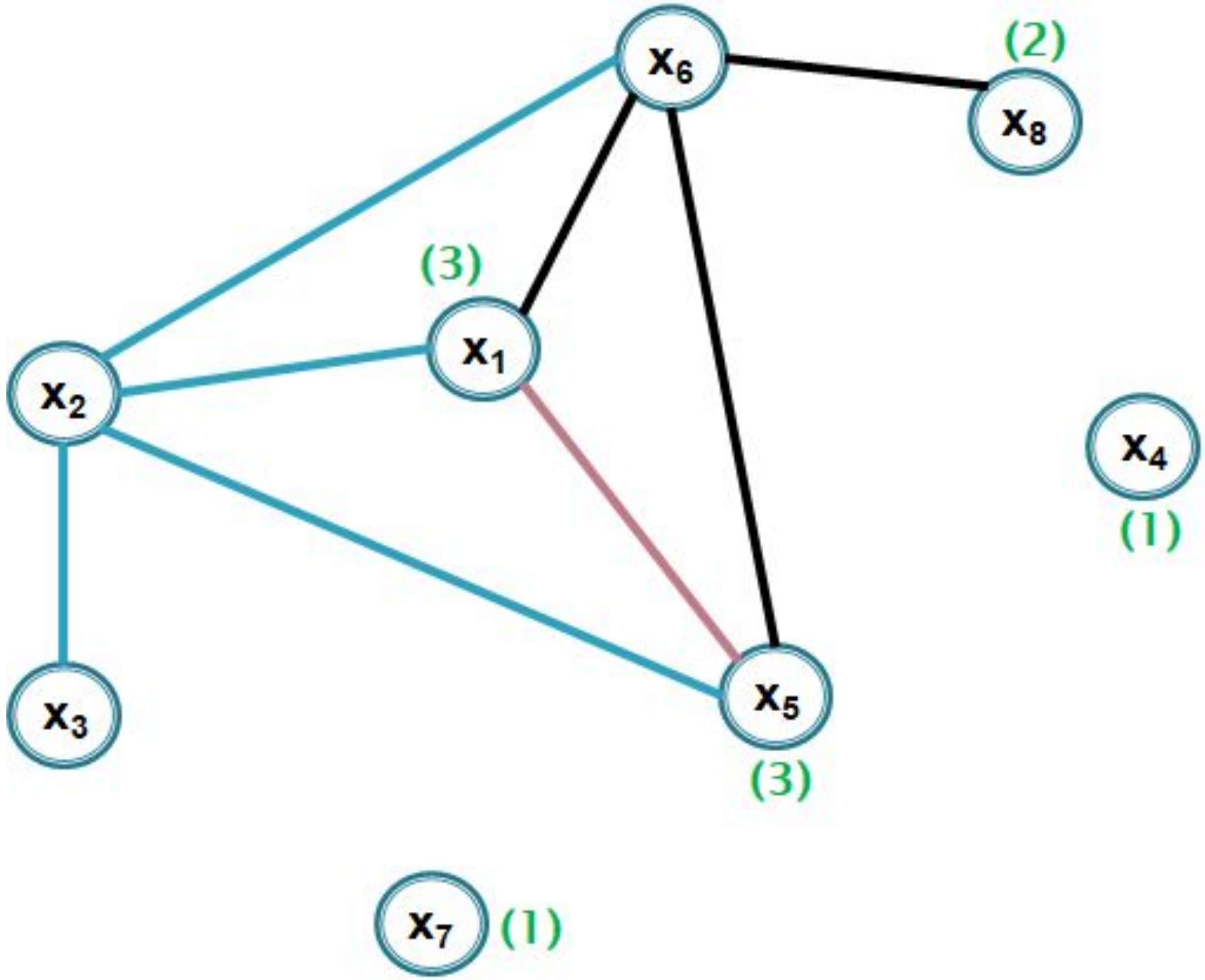
- Muchii construite: x_7x_3
- Secvența rămasă:

$s^{iv}_0 = \{$	0,	1,	1,	0}
etichete vârfuri	x_3	x_4	x_8	x_5

Secvența rămasă ordonată descrescător:

$s'''_0 = \{$	1,	1,	0,	0}
etichete vârfuri	x_4	x_8	x_3	x_5

Exemplu Algoritm



Exemplu Algoritm

Pasul 5.

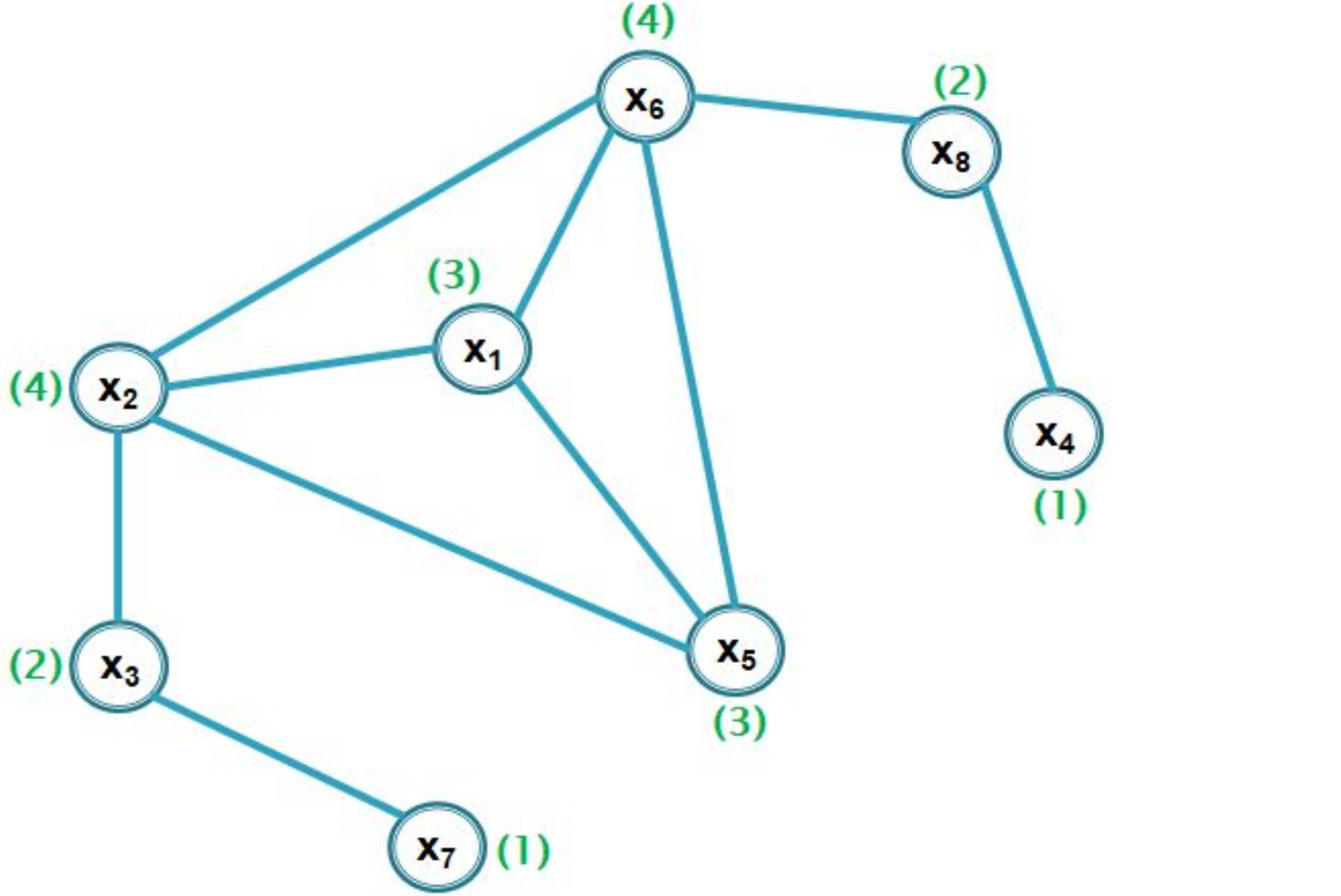
$s^{iv}_0 = \{$	1,	1,	0,	0$\}$
etichete vârfuri	x_4	x_8	x_3	x_5

- Muchii construite: x_4x_8
- Secvența rămasă:

$s^{iv}_0 = \{$	0,	0,	0$\}$
etichete vârfuri	x_8	x_3	x_5

STOP

Exemplu Algoritm



Algoritmul Havel-Hakimi

1. Dacă $d_1+\ldots+d_n$ este impar sau există în s_0 un $d_i>n-1$, atunci
   scrie NU, STOP.

2. cât timp s_0 conține valori nenule execută

   alege d_k **cel mai mare număr** din secvența s_0

   elimină d_k din s_0

   fie $d_{i_1},\ldots,d_{i_{d_k}}$ **cele mai mari d_k numere** din s_0

   pentru $j \in \{i_1,\ldots,i_{d_k}\}$ execută:

   - adaugă la G muchia $x_k x_j$
   - înlocuiește d_j în secvența s_0 cu $\mathbf{d_j - 1}$
   - dacă $d_j - 1 < 0$, atunci scrie NU, STOP.

Observație. Pentru a determina ușor care este cel mai mare număr din secvență și care sunt cele mai mari valori care îi urmează, **este util ca pe parcursul algoritmului secvența s_0 să fie ordonată descrescător.**

Complexitate?

Algoritmul Havel-Hakimi - Corectitudine

Teorema Havel–Hakimi

O secvență de n $\geq$ 2 numere naturale

$$s_0 = \{d_1 \geq \ldots \geq d_n\}$$

cu $d_1 \leq n-1$ este secvența gradelor unui graf neorientat (cu n vârfuri)

$\Leftrightarrow$ secvența

$$s_0' = \{d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n\}$$

este secvența gradelor unui graf neorientat (cu n−1 vârfuri).

Observație: Secvența $s_0' = \{d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n\}$ se obține din s_0 eliminând primul element (adică d_1) și scăzând 1 din primele d_1 elemente rămase – acestea au indicii 2, 3, ..., d_1+1

Algoritmul Havel-Hakimi - Corectitudine

- Cu ajutorul transformării t pe pătrat putem obține pornind de la un graf G toate grafurile cu secvența gradelor s(G)

  (și mulțimea vârfurilor V(G))

Algoritmul Havel-Hakimi - Corectitudine - Demonstrație

Algoritmul Havel-Hakimi - Corectitudine

Teorema Havel–Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?

Algoritmul Havel-Hakimi - Corectitudine

Teorema Havel–Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?

Se poate renunța la această ipoteză $\Rightarrow$

Extindere a Teoremei Havel–Hakimi

Fie $s_0=\{d_1, \ldots, d_n\}$ o secvență de $n \geq 2$ numere naturale cu mai mici sau egale cu $n-1$ și fie $i \in \{1,\ldots,n\}$ fixat. Fie secvența obținută din s_0 astfel:

- eliminăm elementul d_i
- scădem o unitate din primele d_i componente în ordine descrescătoare ale secvenței rămase.

Are loc echivalența:

s_0 este secvența gradelor unui graf neorientat $\Leftrightarrow$

este secvența gradelor unui graf neorientat

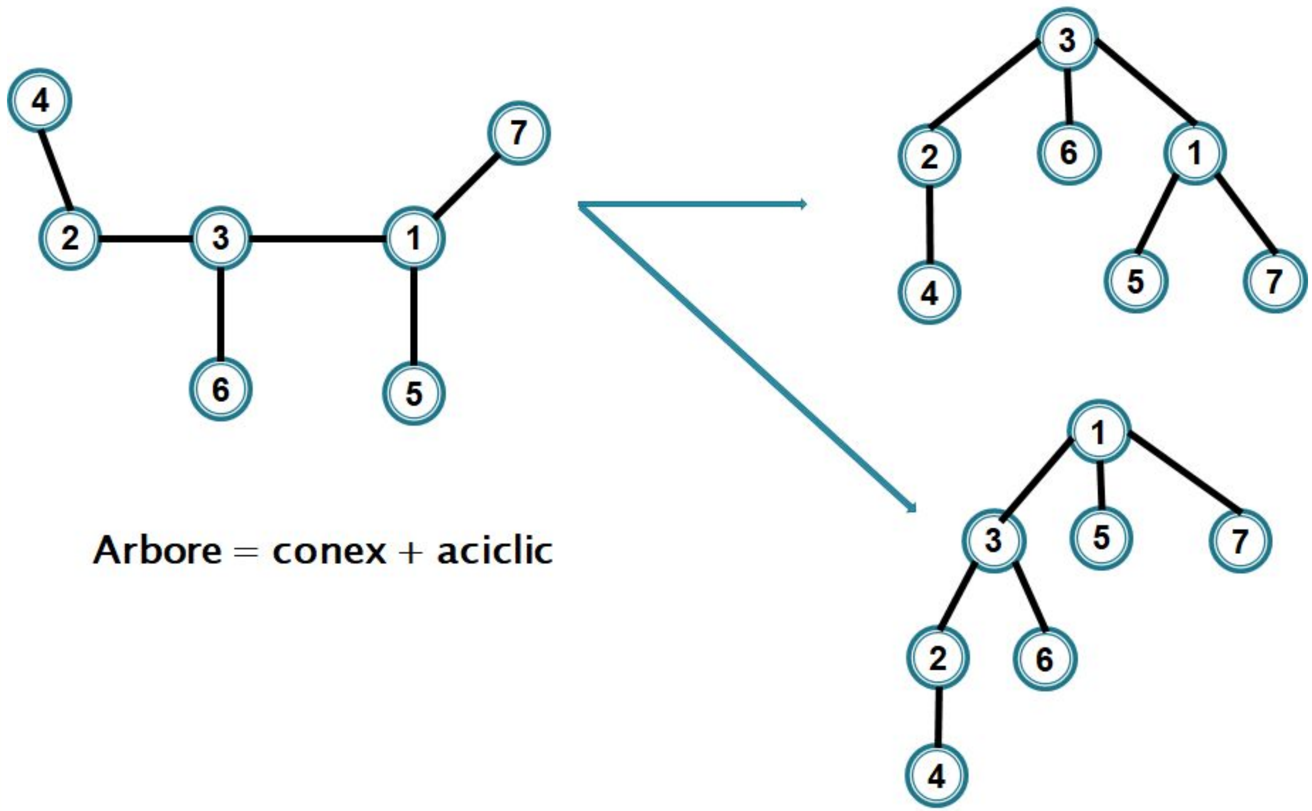
Secventa de grade

Teorema Erdös – Gallai (suplimentar)

O secvență de $n \geq 2$ numere naturale $s_0=\{d_1 \geq \ldots \geq d_n\}$ este secvența gradelor unui graf neorientat $\Leftrightarrow$

- $d_1 + \ldots + d_n$ par si
- $d_1 + \ldots + d_k \leq k(k-1) + \sum_{i=k+1}^{n} \min\{d_i, k\}, \forall 1 \leq k \leq n$

Arbori



Arbore cu rădăcină

Arbori

Leme

1. Orice arbore T cu $n>1$ are cel puțin două vârfuri terminale (de grad 1)

Arbori

Leme

1. Orice arbore T cu n>1 are cel puțin două vârfuri terminale (de grad 1)

Fie P un **lanț elementar maxim** în T

Extremitățile lui P sunt vârfuri terminale, altfel:

- putem extinde lanțul cu o muchie

sau

- se închide un ciclu în T

Arbori

Leme

2. Fie T un arbore cu $n>1$ vârfuri și v un vârf terminal în T.

   Atunci $T - v$ este arbore.

Arbori

Leme

3. Un arbore cu n vârfuri are n−1 muchii.

Arbori

Leme

3. Un arbore cu n vârfuri are n−1 muchii.

Inducție după n

- Dacă T este un arbore cu n vârfuri și v este vârf terminal în T, atunci T − v este arbore cu n−1 vârfuri și $|E(T-v)| = |E(T)|-1$
- **Aplicăm ipoteza de inducție pentru T−v**

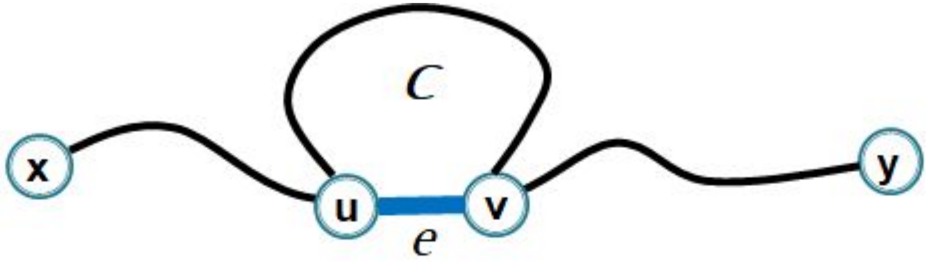
Arbori

Observație

Fie G un graf neorientat conex și C un ciclu in G.

Fie e $\in$ E(C) o muchie din ciclul C.

Atunci G-e este tot un graf conex.



Arbori

Definiții echivalente

Fie T un graf neorientat cu n>1 vârfuri.

Următoarele afirmații sunt echivalente.

1. **T este arbore (conex și aciclic)**
2. **T este conex muchie-minimal**
3. **T este aciclic muchie-maximal**
4. **T este conex și are n-1 muchii**
5. **T este aciclic și are n-1 muchii**
6. **Între oricare două vârfuri din T există un unic lanț elementar.**

Construcția de arbori cu secvența gradelor dată

Fie $s_0 = \{d_1, \dots, d_n\}$.

Condiții necesare pentru ca s_0 să fie secvența gradelor unui arbore?

Construcția de arbori cu secvența gradelor dată

Fie $s_0 = \{d_1, \dots , d_n\}$.

Condiții necesare pentru ca s_0 **să fie secvența gradelor unui arbore?**

$$d_1 + \dots + d_n = 2(n-1)$$

Construcția de arbori cu secvența gradelor dată

Fie $s_0 = \{d_1, \ldots, d_n\}$.

Condiții necesare pentru ca s_0 să fie secvența gradelor unui arbore?

$$d_1 + \ldots + d_n = 2(n-1)$$

Consecințe:

$d_{max}>1$, $d_{min}=1$ pentru n>2

Construcția de arbori cu secvența gradelor dată

Fie $s_0 = \{d_1, \ldots, d_n\}$.

Teorema:

Exista T - un arbore cu *n* noduri astfel încât $S(T)=s_0$ dacă și numai dacă $d_1 + \ldots + d_n = 2(n-1)$

But why?

Construcția de arbori cu secvența gradelor dată

Fie $s_0 = \{d_1, \dots, d_n\}$.

Teorema:

Exista T - un arbore cu *n* noduri astfel încât $S(T)=s_0$ dacă și numai dacă $d_1 + \dots + d_n = 2(n-1)$

Implicația directă - evident

Implicația inversă - inducție

Construcția de arbori cu secvența gradelor dată

Algoritm – Pseudocod

1. Dacă $d_1 + \ldots + d_n \neq 2(n-1)$, atunci scrie NU, STOP.

2. Cât timp s_0 conține valori mai mari decât 1 execută //pentru i=1,n-2

   - alege un număr $\mathbf{d_k} > \mathbf{1}$ și un număr $\mathbf{d_t} = \mathbf{1}$ din secvență s_0
   - adaugă la T muchia $x_k x_t$.
   - elimină d_t din s_0
   - înlocuiește d_k în secvența s_0 cu $\mathbf{d_k} - \mathbf{1}$

3. fie d_k, d_t unicele elemente nenule (egale cu 1) din s_0;
   adaugă la T muchia $x_k x_t$

Construcția de arbori cu secvența gradelor dată

Algoritm 2 - omida (tabla)

Construcția de arbori cu secvența gradelor dată

- reprezentare arbori in memorie - de ce este nevoie?
- algortimii Prufer si Neville
- Teorema lui Cayley
- Teorema lui Moon