Construcția de grafuri cu secvența gradelor dată

Secvenţe de grade

Dată o formulă chimică, există un compus chimic cu această formulă? Dar unul aciclic? Ce structuri poate avea un astfel de compus?

- C_mH_n - poate exista moleculă **aciclică** cu această formulă?

Secvenţe de grade

Din studii empirice, chestionare, analize ⇒ informații despre numărul de interacțiuni ale unui nod

Este realizabilă o rețea de legături între noduri care să respecte numărul de legături?

Dacă da, să se construiască un model de rețea.

Secvenţe de grade

Din studii empirice, chestionare, analize ⇒ informații despre numărul de interacțiuni ale unui nod

Este realizabilă o rețea de legături între noduri care să respecte numărul de legături? Dacă da, să se construiască un model de rețea.

Exemplu: Într-o grupă de studenți, fiecare student este întrebat cu câți colegi a colaborat în timpul anilor de studii. Este realizabilă o rețea de colaborări care să corespundă răspunsurilor lor (sau este posibil ca informațiile adunate să fie incorecte)?

- Studentul 1 - cu 3
- Studentul 2 - cu 3
- Studentul 3 - cu 2
- Studentul 4 - cu 3
- Studentul 5 - cu 2

Secvenţe 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**

Secvenţe de grade

- **Construcția de grafuri cu secvența gradelor dată**

  Aplicații:

  - c**himie** – studiul structurii posibile a unor compuși cu formula chimică dată
  - **proiectare de rețele**
  - **biologie** – rețele metabolice, de interacțiuni între gene/proteine
  - **studii epidemiologice** – în care prin chestionare anonime persoanele declară numărul de persoane cu care au interacționat
  - studii bazate pe simulări de rețele...

Construcția de grafuri neorientate cu secvența gradelor dată.

Algoritmul Havel-Hakimi

Construcția de grafuri 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 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 necesare pentru existența lui G:

- $d_1 + \ldots + d_n$ – număr par
- $d_i \leq n - 1, \forall i$

Construcția de grafuri cu secvența gradelor dată

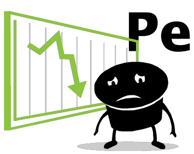
Problemă

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 necesare pentru existența lui G:

- $d_1 + \ldots + d_n$ – număr par
- $d_i \leq n - 1, \forall i$

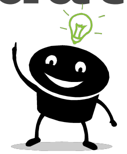


Pentru $s_0 = \{3, 3, 1, 1\}$ - nu există G

$\Rightarrow$ **condițiile nu sunt și suficiente**

… **totusi puteam crea un multigraf**

Construcția de grafuri cu secvența gradelor dată



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 cu secvența gradelor dată

Exemplu algoritm

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

etichete vârfuri 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

- alegem ca vecini următoarele vârfuri cu cele mai mari grade

Construcția de grafuri cu secvența gradelor dată

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 $\Rightarrow$ G**
  - **secvența conține numere negative $\Rightarrow$**

Construcția de grafuri cu secvența gradelor dată

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**
  - **secvența conține numere negative ⇒**

G nu se poate construi prin acest procedeu

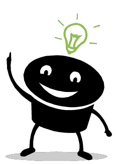
Se poate construi G altfel?

Construcția de grafuri cu secvența gradelor dată

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**
  - **secvența conține numere negative ⇒**

G nu se poate construi prin acest procedeu



Teorema Havel-Hakimi ⇒ NU

⇒ Algoritmul anterior= Algoritmul Havel-Havimi

Exemplu algoritm Havel-Hakimi

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

etichete vârfuri 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 Havel-Hakimi

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

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

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

- alegem ca vecini următoarele vârfuri cu cele mai mari grade

⇒ **ar fi utilă sortarea descrescătoare a elementelor lui s_0**

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

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

Exemplu algoritm Havel-Hakimi

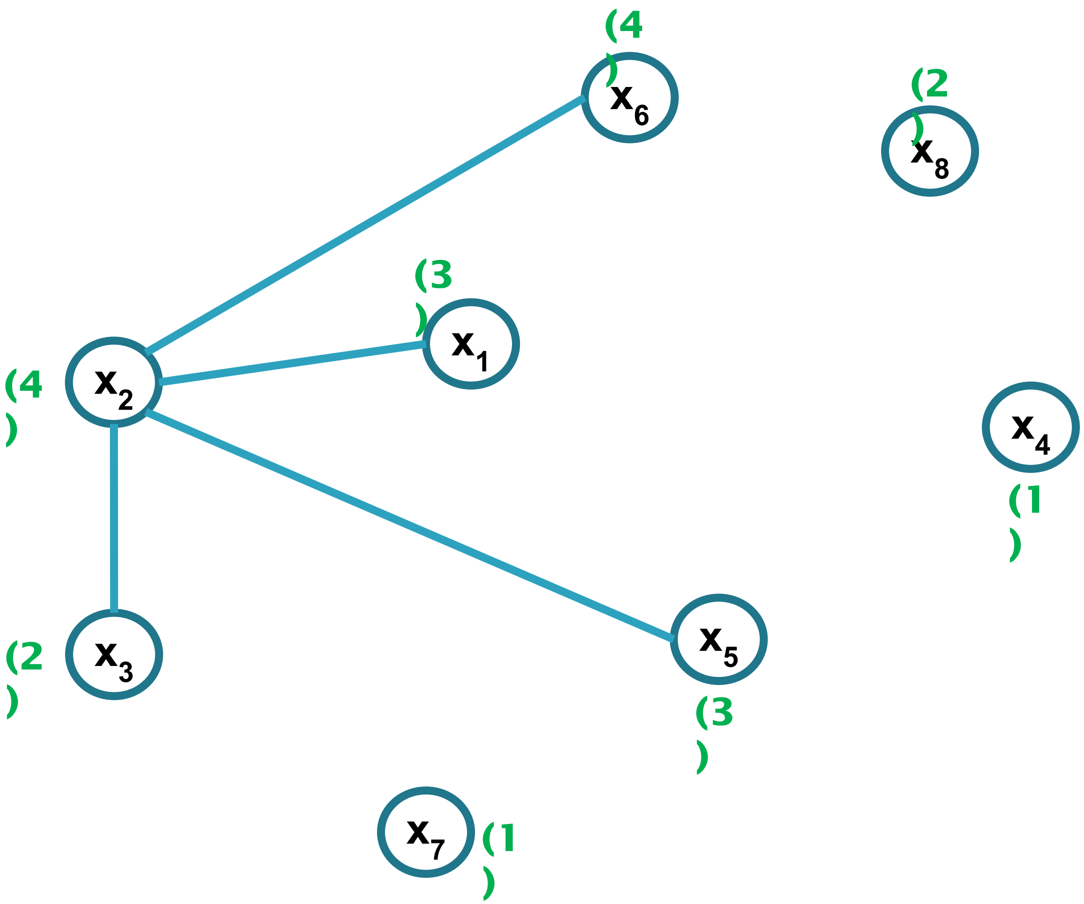
Pasul 1.

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

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 Havel-Hakimi



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 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 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

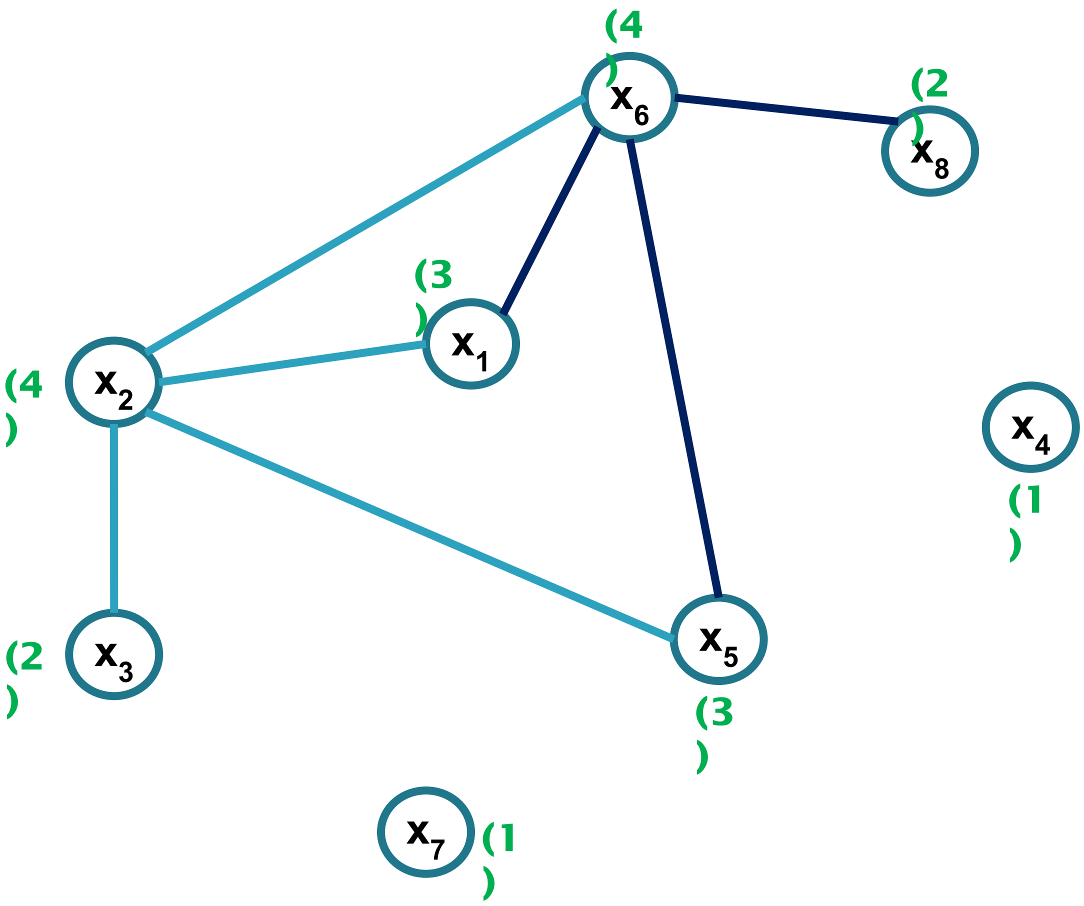
- **Muchii construite:** x_6x_1, x_6x_5, x_6x_8
- **Secvența rămasă:**

$s''_0 = \{$ 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 Havel-Hakimi



Exemplu algoritm Havel-Hakimi

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:**

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

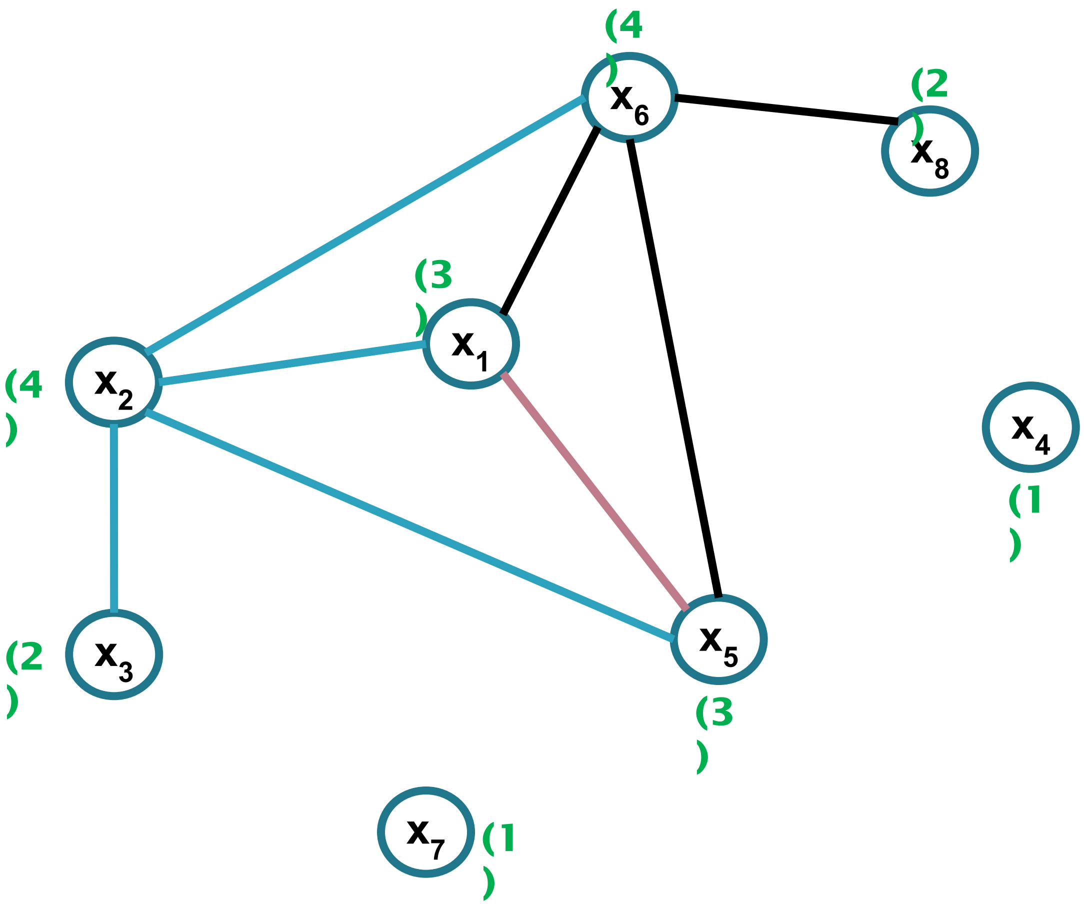
etichete vârfuri x_5 x_8 x_3 x_4 x_7

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 Havel-Hakimi



Exemplu algoritm Havel-Hakimi

Pasul 4.

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

etichete vârfuri $\quad x_7 \quad x_3 \quad x_4 \quad x_8 \quad x_5$

- **Muchii construite:** x_7x_3

- **Secvența rămasă:**

$s^{iv}_0 = \{ \quad 0, \quad 1, \quad 1, \quad 0\}$

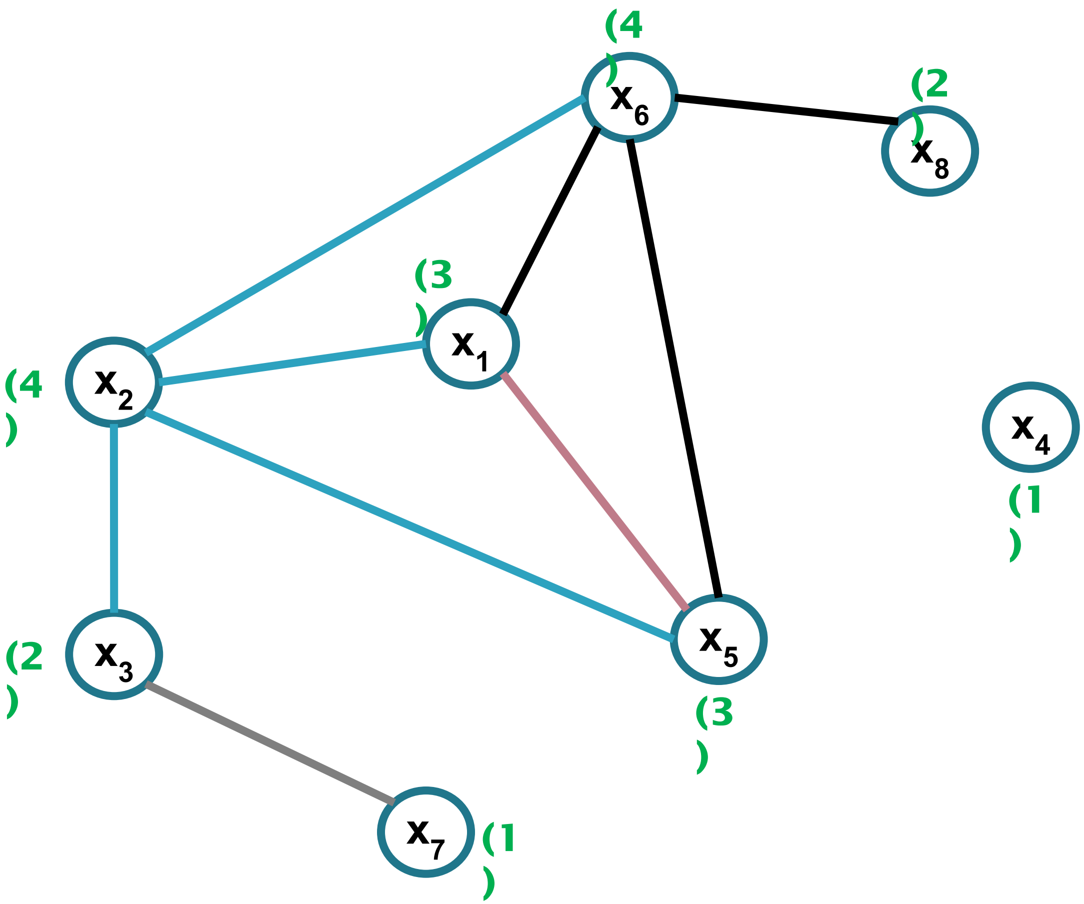
etichete vârfuri $\quad x_3 \quad x_4 \quad x_8 \quad x_5$

Secvența rămasă ordonată descrescător:

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

etichete vârfuri $\quad x_4 \quad x_8 \quad x_3 \quad x_5$

Exemplu algoritm Havel-Hakimi



Exemplu algoritm Havel-Hakimi

Pasul 5.

$s^{iv}_0 = \{$ 1, 1, 0, 0$\}$

etichete vârfuri x_4 x_8 x_3 x_5

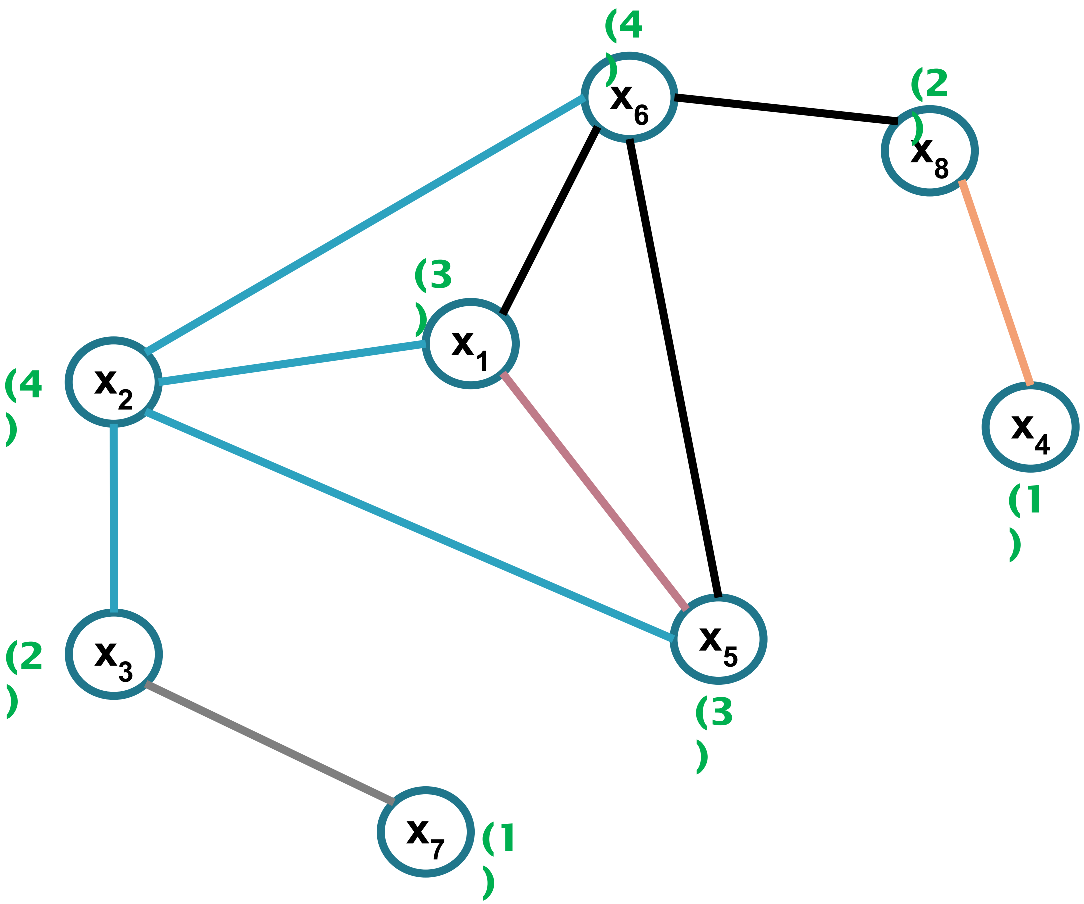
- **Muchii construite:** x_4x_8
- **Secvența rămasă:**

$s^{iv}_0 = \{$ 0, 0, 0$\}$

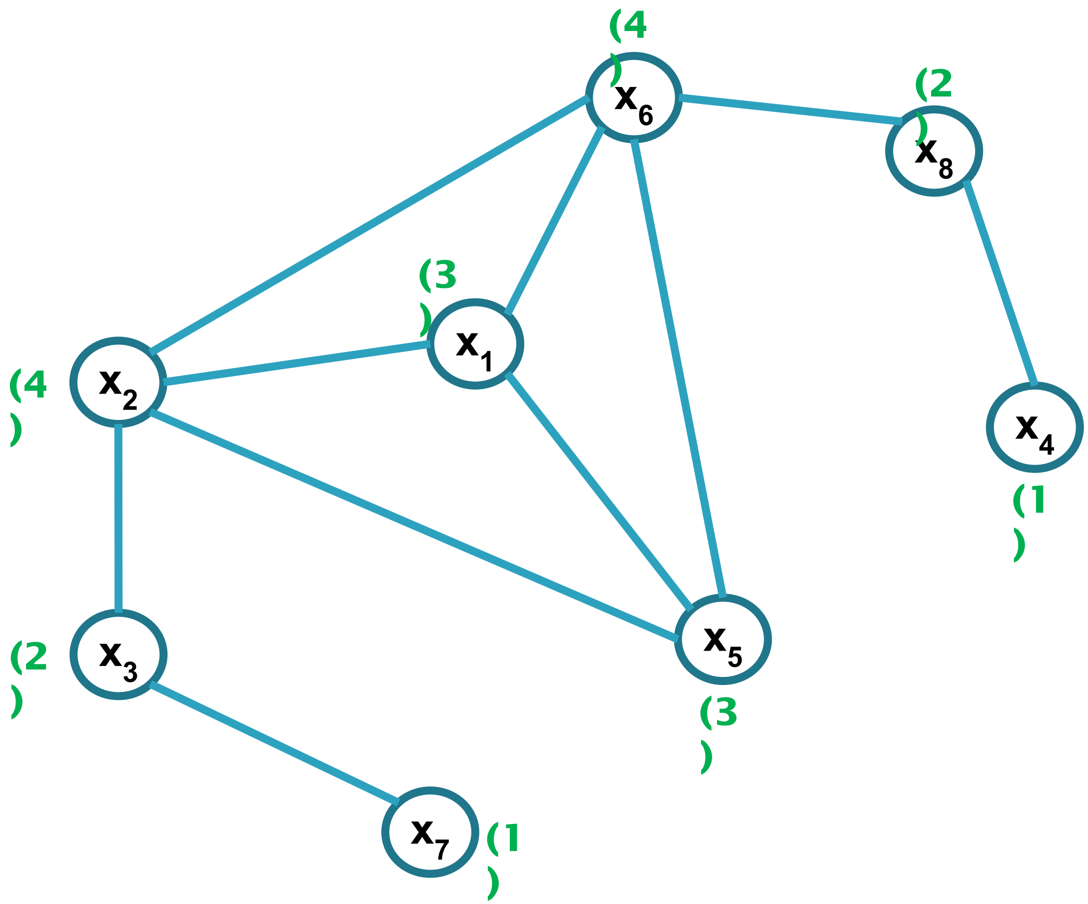
etichete vârfuri x_8 x_3 x_5

STOP

Exemplu algoritm Havel-Hakimi



Exemplu algoritm Havel-Hakimi



Algoritm 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

Algoritm Havel-Hakimi

1. Dacă $d_1+\dots+d_n$ 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 $d_{i_1},\dots,d_{i_{d_k}}$ **cele mai mari d_k numere** din s_0

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

Algoritm 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.

Algoritm 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?

Algoritm 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**.

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

O secvență de n ≥ 2 numere naturale

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

cu d_1 ≤ n-1 este secvența gradelor unui graf neorientat (cu n vârfuri) ⇔ 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).

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

O secvență de n ≥ 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) ⇔ 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).

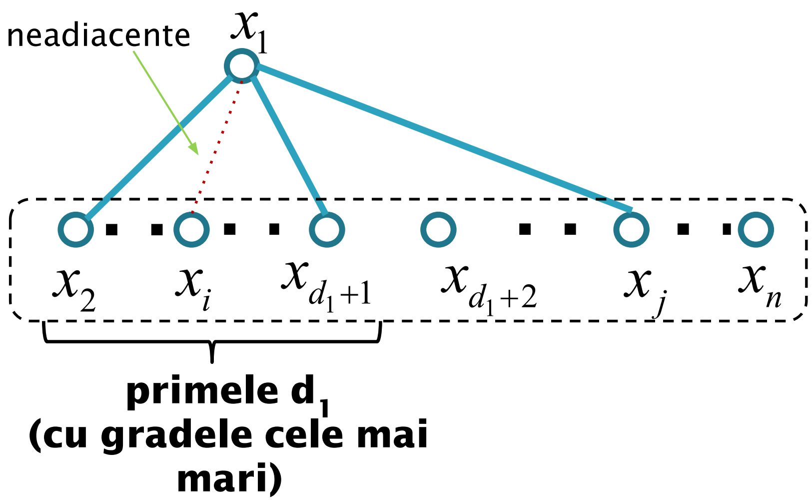
Observație: Secvența $s_0' = \{d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n\}$ se obține din s_0 eliminând primul element (adică d_1) și scăzând 1 din primele d_1 elemente rămase – acestea au indicii 2, 3, ..., d_1+1

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi – Demonstrație

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

$$G,\ s(G) = s_0$$



Algoritm Havel-Hakimi - Corectitudine

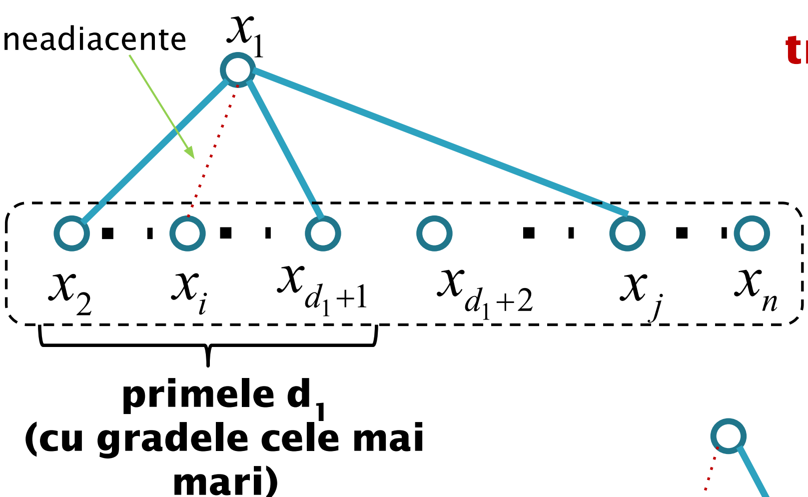
Teorema Havel-Hakimi – Demonstrație

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

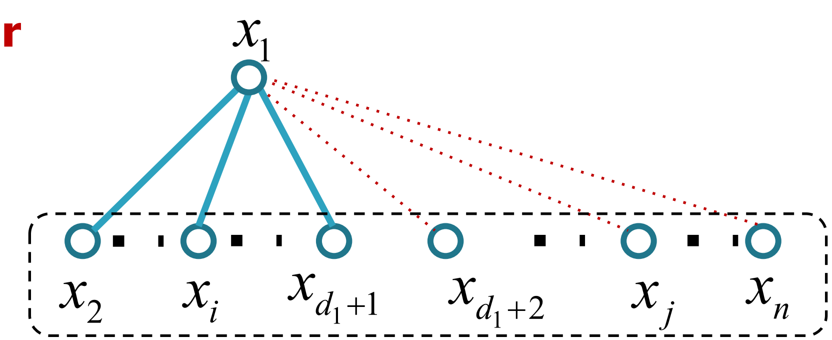
$G,\ s(G) = s_0$

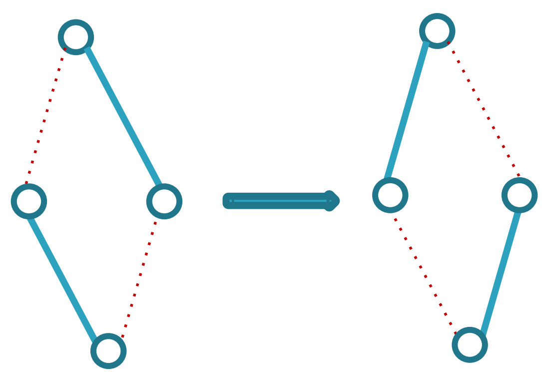
$G^*,\ s(G^*) = s_0$

$N_{G^*}(x_1) = \{x_2, \ldots, x_{d_1+1}\}$



transformar
e t
pe pătrat





Algoritm Havel-Hakimi - Corectitudine

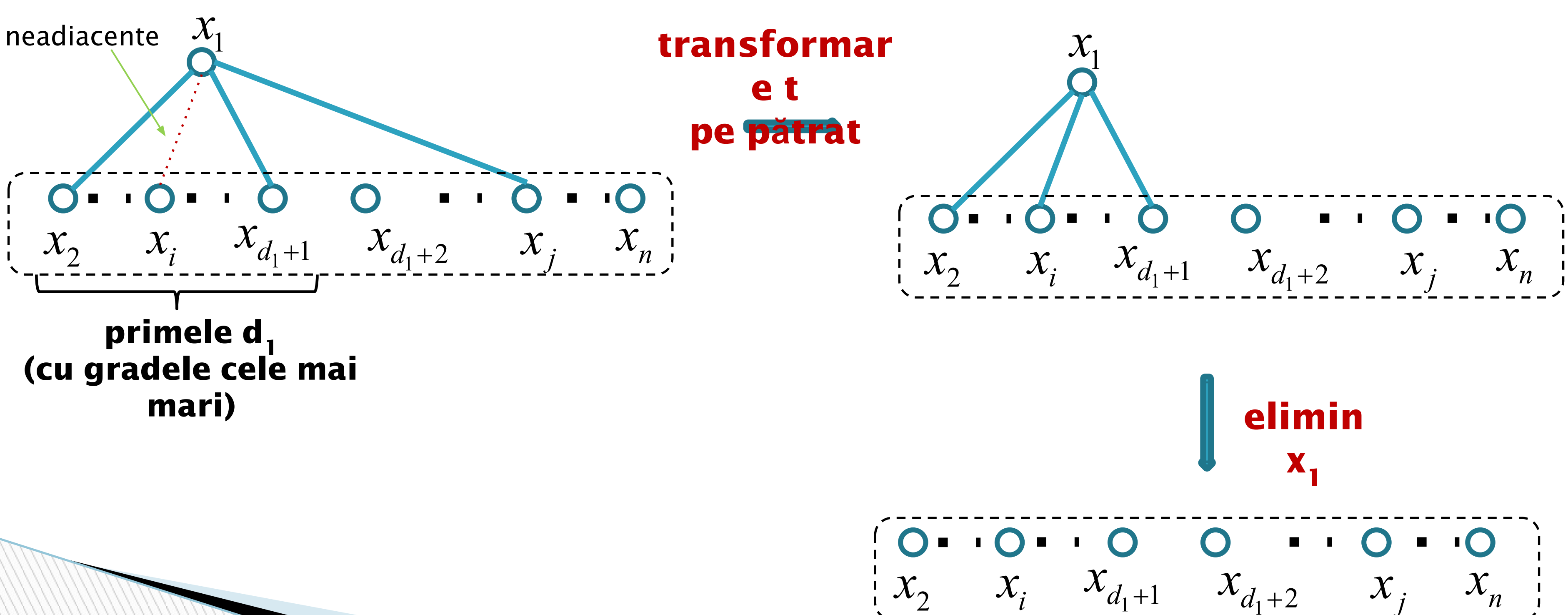
Teorema Havel-Hakimi – Demonstrație

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

$G,\ s(G) = s_0$

$G^*,\ s(G^*) = s_0$

$N_{G^*}(x_1) = \{x_2, \ldots, x_{d_1+1}\}$



$G' = G^* - x_1,\ s(G') = s_0'$

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi – Demonstrație

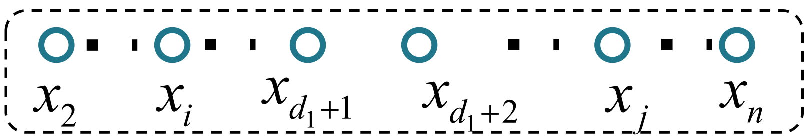
$$s_0 = \{d_1 \geq \ldots \geq d_n\} \quad \Leftarrow \quad s_0' = \{d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n\}$$

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi – Demonstrație

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

Fie G' cu $s(G') = s_0' = \{d_2 - 1, \ldots, d_{d_1+1} - 1, d_{d_1+2}, \ldots, d_n\}$



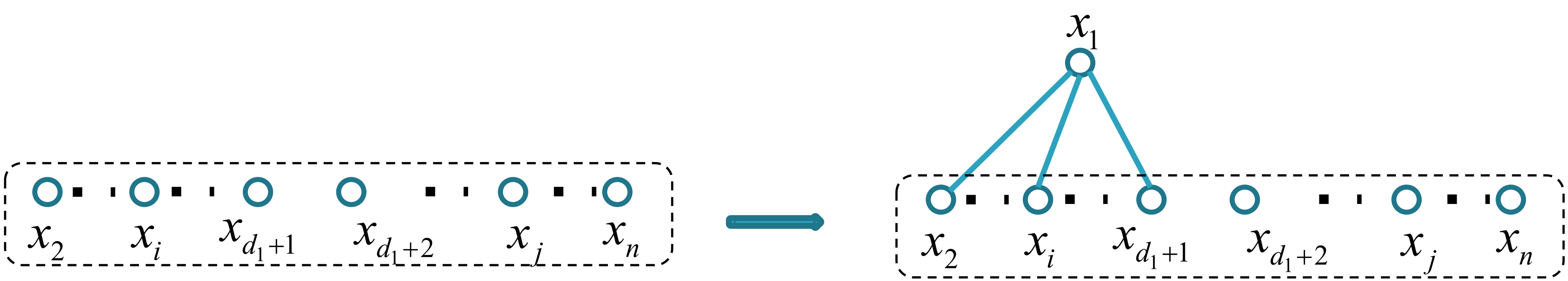
Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi – Demonstrație

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

Fie G' cu $s(G') = s_0'$

$$G: \quad V(G) = V(G') \cup \{x_1\}$$
$$E(G) = E(G') \cup \{x_1x_2, \ldots, x_1x_{d_1+1}\}$$



adăugăm un vârf x_1 pe care îl unim cu $x_2, \ldots, x_{d_1+1}$

Avem $s(G) = s_0$.

Algoritm Havel-Hakimi - Corectitudine

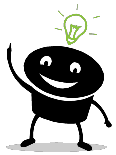
Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?



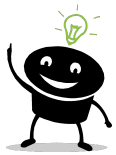
Se poate renunța la această ipoteză ⇒

<u>Extindere</u> a Teoremei Havel-Hakimi

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?



Se poate renunța la această ipoteză ⇒

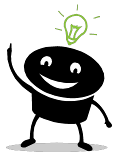
Extindere a Teoremei Havel-Hakimi

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 $s_0^{(i)}$ secvența obținută din s_0 astfel:

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?



Se poate renunța la această ipoteză ⇒

Extindere a Teoremei Havel-Hakimi

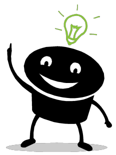
Fie $s_0=\{d_1, ..., d_n\}$ o secvență de $n\geq 2$ numere naturale cu mai mici sau egale cu n-1 și fie $i \in \{1,...,n\}$ fixat. Fie $s_0^{(i)}$ secvența obținută din s_0 astfel:

- eliminăm elementul d_i

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?



Se poate renunța la această ipoteză ⇒

<u>Extindere</u> a Teoremei Havel-Hakimi

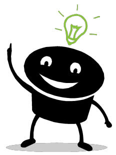
Fie $s_0=\{d_1, \ldots, d_n\}$ o secvență de $n\geq 2$ numere naturale cu mai mici sau egale cu n-1 și fie $i\in \{1,\ldots,n\}$ fixat. Fie $s_0^{(i)}$ secvența obținută din s_0 astfel:

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

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?



Se poate renunța la această ipoteză ⇒

Extindere a Teoremei Havel-Hakimi

Fie $s_0=\{d_1, \ldots, d_n\}$ o secvență de $n\geq2$ numere naturale cu mai mici sau egale cu n-1 și fie $i\in \{1,\ldots,n\}$ fixat. Fie $s_0^{(i)}$ 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 ⇔

$s_0^{(i)}$ este secvența gradelor unui graf neorientat

Algoritm Havel-Hakimi - Corectitudine

Teorema Havel-Hakimi

Unde intervine în demonstrație faptul că d_1 este maxim?

Se poate renunța la această ipoteză ⇒

Extindere a Algoritmului Havel-Hakimi

La un pas vârful poate fi ales arbitrar (nu neapărat cel corespunzător elementului maxim).

Se păstrează însă criteriul de alegere al vecinilor (cu gradele cele mai mari)

Construcția de grafuri cu secvența gradelor dată

- **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))

Construcția de grafuri cu secvența gradelor dată

- **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))

- Mai exact, ar loc următorul rezultat (exercițiu):

  Fie G_1 și G_2 două grafuri neorientate cu mulțimea vârfurilor $V=\{1,\dots,n\}$.

  Atunci $s(G_1)=s(G_2) \Leftrightarrow$ există un șir de transformări t de interschimbare pe pătrat prin care se poate obține graful G_2 din G_1.

Construcția de grafuri cu secvența gradelor dată

Teorema Erdös - Gallai (suplimentar)

O secvență de n ≥ 2 numere naturale $\mathbf{s}_0=\{d_1 \geq \ldots \geq d_n\}$ este secvența gradelor unui graf neorientat ⇔

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