# Calcul numeric

## Procesare de semnal pe grafuri

Paul Irofti
Andrei Pătrașcu
Cristian Rusu

Departmentul de Informatică

Facultatea de Matematică și Informatică

Universitatea din București

Email: prenume.nume@fmi.unibuc.ro

# Cuprins

- Ce am făcut data trecută: descompunerea valorilor proprii (în special cazul simetric)
- Motivație pentru procesarea semnalelor pe grafuri
- Definiții
- Algoritmi
- Exemple
- Aplicații

# Descompunerea valorilor proprii (cazul general)

Pentru $\mathbf{A} \in \mathbb{R}^{n \times n}$.
Dacă avem $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ pentru $i = 1, \ldots, n$ atunci avem defapt

$$\mathbf{A} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n \end{bmatrix} \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix} \tag{1}$$

adică

$$\mathbf{AV} = \mathbf{VD} \text{ sau } \mathbf{A} = \mathbf{VDV}^{-1}. \tag{2}$$

# Descompunerea valorilor proprii (cazul simetric)

Pentru $\mathbf{A} \in \mathbb{R}^{n\times n}$ și $\mathbf{A} = \mathbf{A}^T$.
Dacă avem $\mathbf{A}\mathbf{v}_i = \lambda_i \mathbf{v}_i$ pentru $i = 1, \ldots, n$ atunci avem defapt

$$\mathbf{A}\begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n\end{bmatrix} = \begin{bmatrix}\mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_n\end{bmatrix}\begin{bmatrix}\lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n\end{bmatrix} \tag{3}$$

adică

$$\mathbf{AV} = \mathbf{VD} \text{ sau } \mathbf{A} = \mathbf{VDV}^T. \tag{4}$$

(adică, $\mathbf{V}^{-1} = \mathbf{V}^T$)

# Motivația: procesarea semnalelor pe grafuri

Foarte multe situații se modelează matematic sub formă de grafuri

- page rank
- interacțiunile pe rețelele sociale
- rețele de drumuri
- rețele electrice
- sisteme complexe în fizică/chimie
- structuri moleculare
- procesare de imagini
- rețele neuronale convoluționale (pe grafuri)

# Definiții: grafuri

ce este un graf?

- un set de vârfuri $\mathcal{V} \in \{1, \ldots, n\}$ pentru un graf cu $N$ vârfuri
- un set de muchii între vârfurile grafului $\mathcal{M} \in \mathcal{V} \times \mathcal{V}$
- avem graful $\mathcal{G} = (\mathcal{V}, \mathcal{M})$

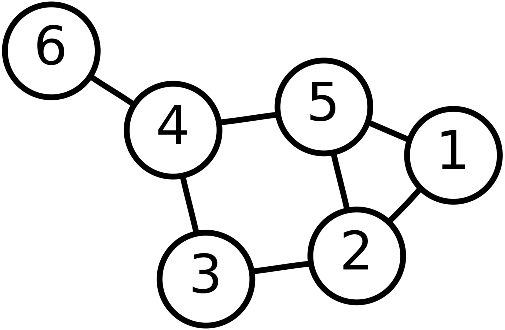



Figura: Graf neorientat, neponderat (sursa: wikipedia)

# Definiții: grafuri

ce este un graf?

- un set de vârfuri $\mathcal{V} \in \{1, \ldots, n\}$ pentru un graf cu $N$ vârfuri
- un set de muchii între vârfurile grafului $\mathcal{M} \in \mathcal{V} \times \mathcal{V}$
- avem graful $\mathcal{G} = (\mathcal{V}, \mathcal{M})$

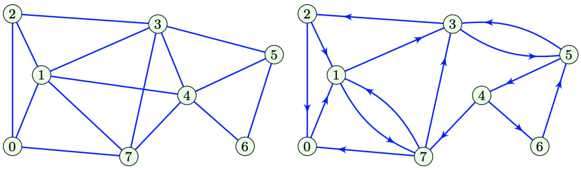



Figura: Graf neorientat vs. orientat (sursa: Springer-Nature)

# Definiții: grafuri

ce este un graf?

- un set de vârfuri $\mathcal{V} \in \{1, \dots, n\}$ pentru un graf cu $N$ vârfuri
- un set de muchii între vârfurile grafului $\cancel{\mathcal{M} \in \mathcal{V} \times \mathcal{V}}$
  $m : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$
- avem graful $\mathcal{G} = (\mathcal{V}, \mathcal{M})$

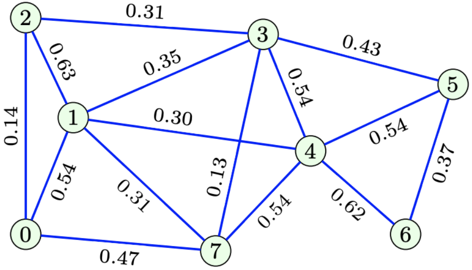



Figura: Graf ponderat (sursa: Springer-Nature)

# Definiții: grafuri

ce este un graf?

- până acum ați văzut multe grafuri în care nodurile reprezintă o conexiune abstractă
- până acum muchiile și valorile lor erau importante (de exemplu: algoritmi drum de cost minim etc.)
- în acest curs, în fiecare nod avem o valoare (nu doar index-ul nodului):
  - temperatura dintr-un oraș
  - nivelul apei dintr-o rețea de apă
  - numărul de persoane dintr-o comunitate
  - traficul cumulat într-un nod
  - etc.

# Definiții: grafuri

până acum aveați situația aceasta:

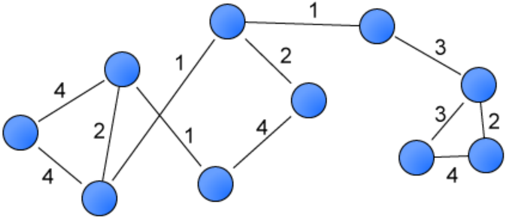



Figura: Graf neorientat, ponderat (sursa: wikipedia)

# Definiții: grafuri

astăzi avem un semnal pe graf (în nodurile lui) astfel:

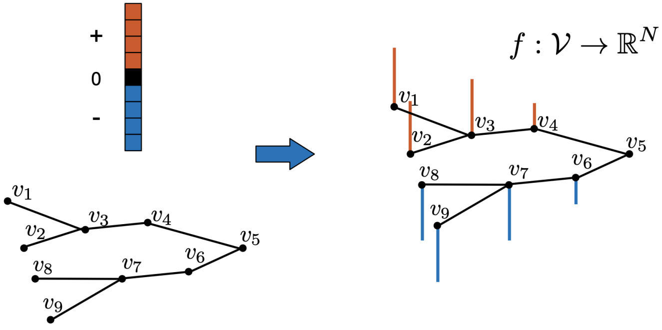



Figura: Graf neorientat, ponderat, dar cu semnal (sursa: MIT)

# Definiții: exemplu

o rețea de senzori

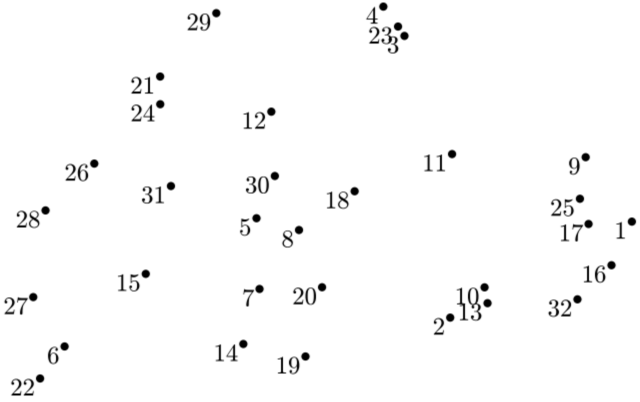



Figura: Senzori amplasați în spațiu (sursa: Springer-Nature)

# Definiții: exemplu

o rețea de senzori și vecinii unui senzor

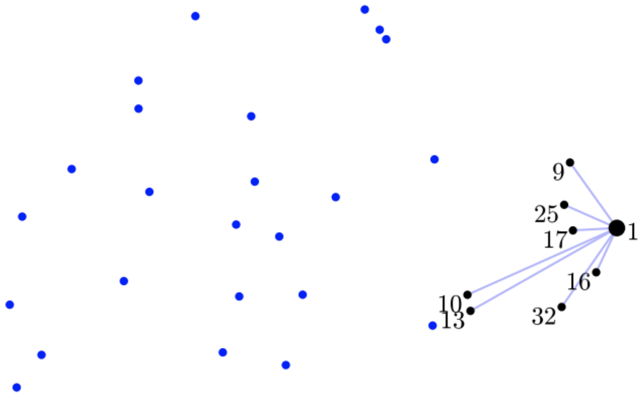



Figura: Comunicarea între senzori (sursa: Springer-Nature)

# Definiții: exemplu

o rețea de senzori și comunicarea dintre ei

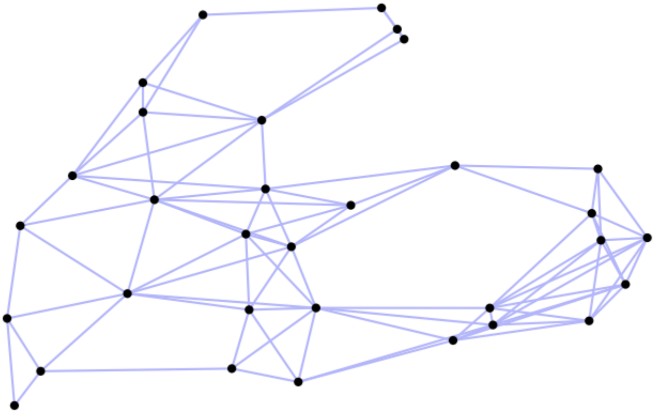


Figura: Structura de graf a senzorilor (sursa: Springer-Nature)

# Definiții: exemplu

o rețea de senzori

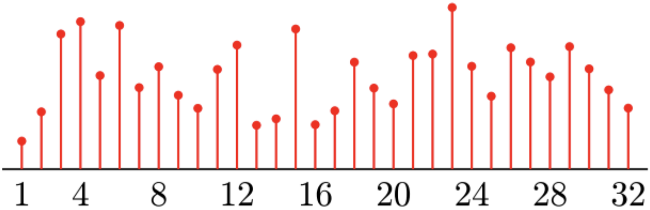



Figura: Măsurători de la senzori (sursa: Springer-Nature)

# Definiții: exemplu

o rețea de senzori: viziunea de ansamblu (graf + măsurători)

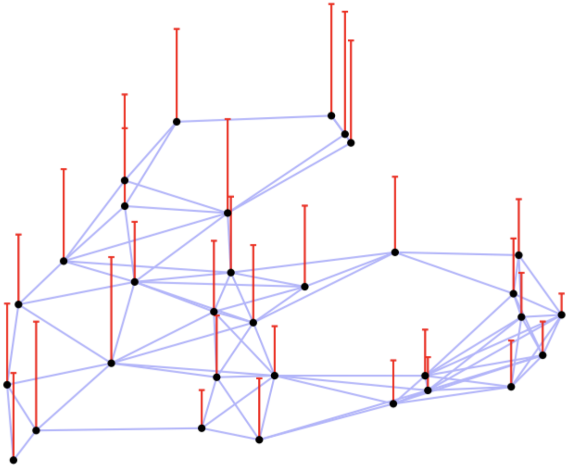
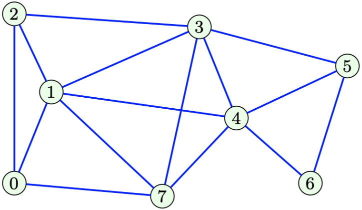
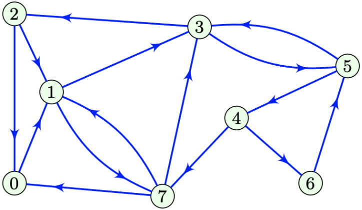


Figura: Măsurători de la senzori (sursa: Springer-Nature)

# Definiții: grafuri

care este legătura cu matricele? matricele de adiacență **A**!

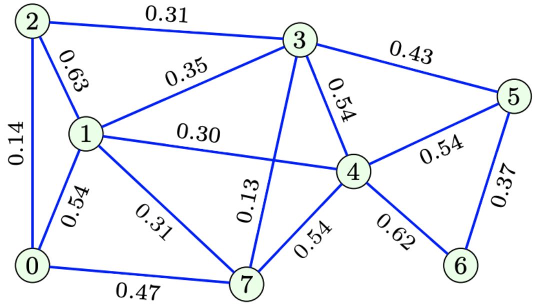



$$
\mathbf{A} = \begin{array}{c}
0\\1\\2\\3\\4\\5\\6\\7\\ \\
\end{array}
\begin{array}{c}
\begin{bmatrix}
0 & 1 & 1 & 0 & 0 & 0 & 0 & 1\\
1 & 0 & 1 & 1 & 1 & 0 & 0 & 1\\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0\\
0 & 1 & 1 & 0 & 1 & 1 & 0 & 1\\
0 & 1 & 0 & 1 & 0 & 1 & 1 & 1\\
0 & 0 & 0 & 1 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 1 & 1 & 0 & 0\\
1 & 1 & 0 & 1 & 1 & 0 & 0 & 0
\end{bmatrix}\\
\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix}
\end{array},
\quad
\mathbf{A} = \begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1\\
1 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1\\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\
1 & 1 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

# Definiții: grafuri

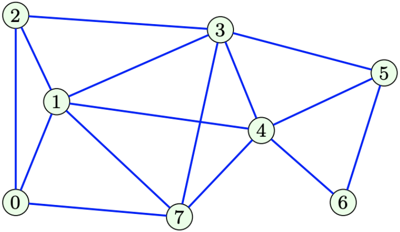
care este legătura cu matricele? matricele de adiacență **A**!




$$\mathbf{W} = \begin{bmatrix} 0 & 0.54 & 0.14 & 0 & 0 & 0 & 0 & 0.47 \\ 0.54 & 0 & 0.63 & 0.35 & 0.30 & 0 & 0 & 0.31 \\ 0.14 & 0.63 & 0 & 0.31 & 0 & 0 & 0 & 0 \\ 0 & 0.35 & 0.31 & 0 & 0.54 & 0.43 & 0 & 0.13 \\ 0 & 0.30 & 0 & 0.54 & 0 & 0.54 & 0.62 & 0.54 \\ 0 & 0 & 0 & 0.43 & 0.54 & 0 & 0.37 & 0 \\ 0 & 0 & 0 & 0 & 0.62 & 0.37 & 0 & 0 \\ 0.47 & 0.31 & 0 & 0.13 & 0.54 & 0 & 0 & 0 \end{bmatrix}$$

de ce este importantă matricea de adiacență?

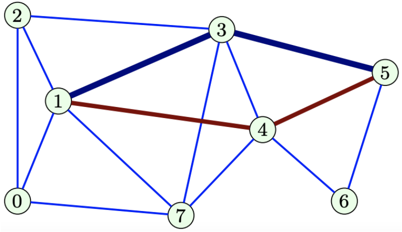
ea (și puterile ei) oferă informații despre drumurile din graf




$$\mathbf{A}^2 = \begin{bmatrix} 3 & 2 & 1 & 3 & \mathbf{2} & 0 & 0 & 1 \\ 2 & 5 & 2 & 3 & 2 & 2 & 1 & 3 \\ 1 & 2 & 3 & 1 & 2 & 1 & 0 & 3 \\ 3 & 3 & 1 & 5 & 3 & 1 & 2 & 2 \\ 2 & 2 & 2 & 3 & 5 & 2 & 1 & 2 \\ 0 & 2 & 1 & 1 & 2 & 3 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 & 1 & 2 & 1 \\ 1 & 3 & 3 & 2 & 2 & 2 & 1 & 4 \end{bmatrix}$$

de ce este importantă matricea de adiacență?
ea (și puterile ei) oferă informații despre drumurile din graf

$$\mathbf{A}^2 = \begin{bmatrix} 3 & 2 & 1 & 3 & \mathbf{2} & 0 & 0 & 1 \\ 2 & 5 & 2 & 3 & 2 & 2 & 1 & 3 \\ 1 & 2 & 3 & 1 & 2 & 1 & 0 & 3 \\ 3 & 3 & 1 & 5 & 3 & 1 & 2 & 2 \\ 2 & 2 & 2 & 3 & 5 & 2 & 1 & 2 \\ 0 & 2 & 1 & 1 & 2 & 3 & 1 & 2 \\ 0 & 1 & 0 & 2 & 1 & 1 & 2 & 1 \\ 1 & 3 & 3 & 2 & 2 & 2 & 1 & 4 \end{bmatrix}$$



# Definiții: grafuri

- deci, $\mathbf{B}_K = \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^K$ ne dă toate posibilitățile de a ajunge de la un nod la altul în maxim $K$ pași prin graf
- această valoare este un indicator pentru cât de strânsă este legătura între oricare două noduri din graf
- deci, va fi important să adunăm puteri ale matricei de adiacență
- dar cum apar în această poveste valorile/vectorii proprii ale matricei $\mathbf{A}$?
  - un răspuns este evident
  - un răspuns este mai puțin evident

# Definiții: grafuri

- deci, $\mathbf{B}_K = \mathbf{A} + \mathbf{A}^2 + \cdots + \mathbf{A}^K$ ne dă toate posibilitățile de a ajunge de la un nod la altul în maxim $K$ pași prin graf
- această valoare este un indicator pentru cât de strânsă este legătura între oricare două noduri din graf
- deci, va fi important să adunăm puteri ale matricei de adiacență
- dar cum apar în această poveste valorile/vectorii proprii ale matricei $\mathbf{A}$?
  - un răspuns este evident: $\mathbf{A}^c = \mathbf{U}\mathbf{D}^c\mathbf{U}^{-1}$ și apoi $\mathbf{B}_K = \mathbf{U}(\mathbf{D} + \mathbf{D}^2 + \cdots + \mathbf{D}^K)\mathbf{U}^{-1}$
  - un răspuns este mai puțin evident

# Definiții: grafuri

- dar cum apar în această poveste valorile/vectorii proprii ale matricei $\mathbf{A}$?
  - un răspuns este mai puțin evident
  - haideți să mai vedem o dată un exemplu de mai sus în care am numerotat nodurile
  - orice facem, trebuie să fie invariant la ordonarea nodurilor
  - adică, un graf care are matricea de adiacență $\mathbf{A}$ este echivalent cu un alt graf care are matricea de adiacență $\mathbf{PAP}^T$ unde $\mathbf{P}$ este o matrice de permutare

$$\mathbf{P} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

# Definiții: grafuri

- dar cum apar în această poveste valorile/vectorii proprii ale matricei $\mathbf{A}$?
  - un răspuns este mai puțin evident
  - haideți să mai vedem o dată un exemplu de mai sus în care am numerotat nodurile
  - orice facem, trebuie să fie invariant la ordonarea nodurilor
  - adică, un graf care are matricea de adiacență $\mathbf{A}$ este echivalent cu un alt graf care are matricea de adiacență $\mathbf{PAP}^T$ unde $\mathbf{P}$ este o matrice de permutare
  - avem $\mathbf{A} = \mathbf{UDU}^{-1}$ și atunci avem

$$\begin{aligned}\mathbf{PAP}^T &= \mathbf{PUDU}^{-1}\mathbf{P}^T \\ &= (\mathbf{PU})\mathbf{D}(\mathbf{PU})^{-1} \\ &= \mathbf{U}_{\text{new}}\mathbf{D}\mathbf{U}_{\text{new}}^{-1}\end{aligned}$$

  - adică, dacă permutăm nodurile doar permutăm defapt vectorii proprii ai matricei de adiacență

# Definiții: grafuri

pentru noi, va fi foarte important Laplacianul grafului $\mathbf{L} = \mathbf{G} - \mathbf{A}$ (matricea de grad minus matricea de adiacență)

$$\mathbf{L} = \begin{bmatrix} 3 & -1 & -1 & 0 & 0 & 0 & 0 & -1 \\ -1 & 5 & -1 & -1 & -1 & 0 & 0 & -1 \\ -1 & -1 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 5 & -1 & -1 & 0 & -1 \\ 0 & -1 & 0 & -1 & 5 & -1 & -1 & -1 \\ 0 & 0 & 0 & -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & -1 & -1 & 0 & 0 & 4 \end{bmatrix} \tag{5}$$

# Definiții: grafuri

proprietăți ale Laplacianului

- are dimensiune $n \times n$
- o valoare proprie este 0
- vectorul propriu pentru valoarea proprie 0 este $\mathbf{1}_{n\times 1}$
- este pozitiv definită (toate valorile proprii sunt pozitive)
  - se poate aplica factorizarea Cholesky
  - o matrice pozitiv definita are un radical (o operație folositoare în multe situații)
- vom folosi această matrice pentru a defini frecvența pe grafuri

# Definiții: aplicație

- noțiunea de frecvență pentru semnale obișnuite

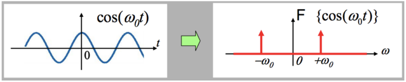
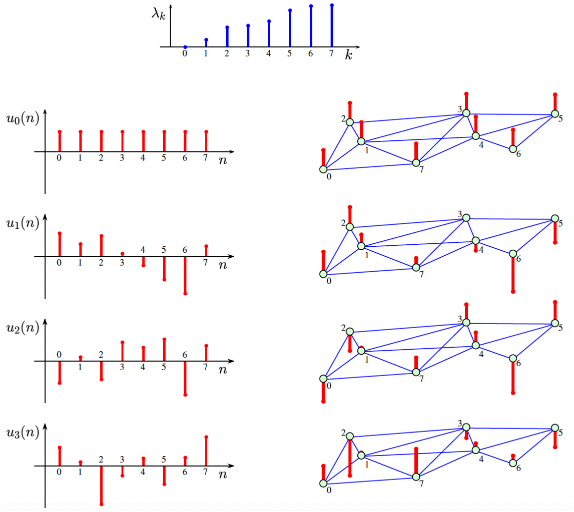



Figura: Descompunerea în frecvență (sursa:http://www.physik.uni-kl.de)

- în cazul nostru:
  - care sunt semnalele de baza pe grafuri? vectorii proprii ai lui **L**!
  - cât de mult contează fiecară bază? valorile proprii ale lui **L**!
  - $\mathbf{U}$ din $\mathbf{L} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ se numește transformata Fourier pe graf

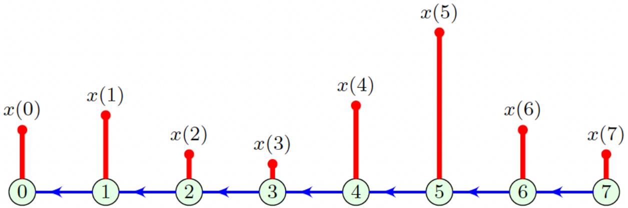
# Definiții: aplicație

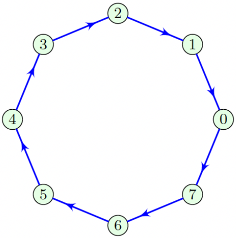
pentru graful precedent, frecvențele și bazele (primele 4) sunt

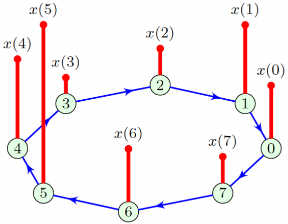


# Definiții: aplicație

de ce se numește **U** transformata Fourier pe graf? când graful este circular în timp atunci **U** este chiar **F** (matricea Fourier)!!!

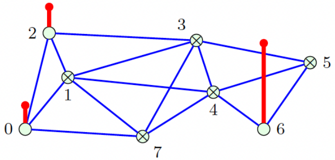


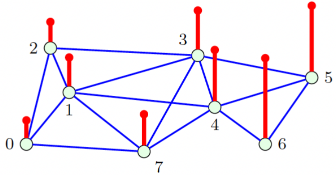



# Definiții: aplicație

o aplicație: dintr-un graf, ni se dau doar câteva valori din noduri iar pe restul trebuie noi să le estimăm




$$\begin{bmatrix} x(0) \\ x(2) \\ x(6) \end{bmatrix} = \begin{bmatrix} u_0(0) & u_1(0) \\ u_0(2) & u_1(2) \\ u_0(6) & u_1(6) \end{bmatrix} \begin{bmatrix} X(0) \\ X(1) \end{bmatrix}$$

$$\mathbf{x} = \mathbf{U}\,\mathbf{X}$$

# Final

- acest domeniu este proaspăt, se face în continuare multă cercetare
- multe principii din timp (serii de timp) sunt extinse la nivel de grafuri
- teorie și aplicații peste ce ați învățat voi la cursuri de structuri de date și algoritmi
- la laborator veți face aplicații
- notițele de curs sunt bazate pe două surse majore:
  - Ljubisa Stankovic, Milos Dakovic, and Ervin Sejdic, Introduction to Graph Signal Processing, Springer-Nature, 2019
  - Xiaowen Dong, Graph signal processing: Concepts, tools and applications in neuroscience, 2019