Drumuri minime în grafuri ponderate

Aplicaţii

- Dată o hartă rutieră, să se determine
  - un drum minim între două orașe date
  - câte un drum minim între oricare două orașe de pe hartă

Aplicaţii

- **Determinarea de drumuri minime/distanţe – numeroase aplicaţii**
  - probleme de rutare
  - robotică
  - procesarea imaginilor
  - strategii jocuri
  - probleme de planificare (drumuri critice)

Cadru

Fie:

- G – graf **orientat** ponderat
- P – drum

$$\mathbf{w(P)} = \sum_{\mathbf{e} \in E(P)} \mathbf{w(e)}$$

costul/ponderea/lungimea drumului P

Cadru

Fie:

- G – graf orientat ponderat
- Presupunem că G nu conține circuite de pondere negativă

Cadru

- Fie $s, v \in V$, $s \neq v$.
- **Distanţa de la s la v**

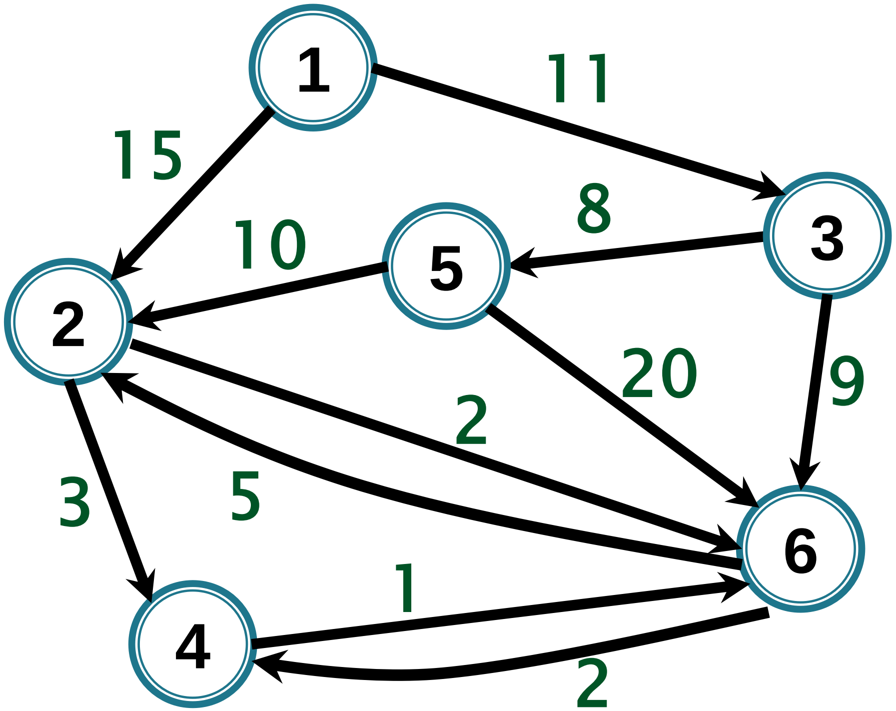
$$\delta_G(s, v) = \min\{ w(P) \mid P \text{ este drum de la } s \text{ la } v\}$$

Cadru

- Fie $s, v \in V$, $s \neq v$.
- **Distanţa de la s la v**

  $$\delta_G(s, v) = \min\{ w(P) \mid P \text{ este drum de la } s \text{ la } v\}$$

  - $\delta_G(s, s) = 0$
  - **Convenție.** $\min \varnothing = \infty$
- Un drum P de la s la v se numește **drum minim de la s la v** dacă $w(P) = \delta_G(s, v)$


1
2
3
4
5
6
15
11
10
8
20
9
2
5
3
1
2

Tipuri de probleme de drumuri minime

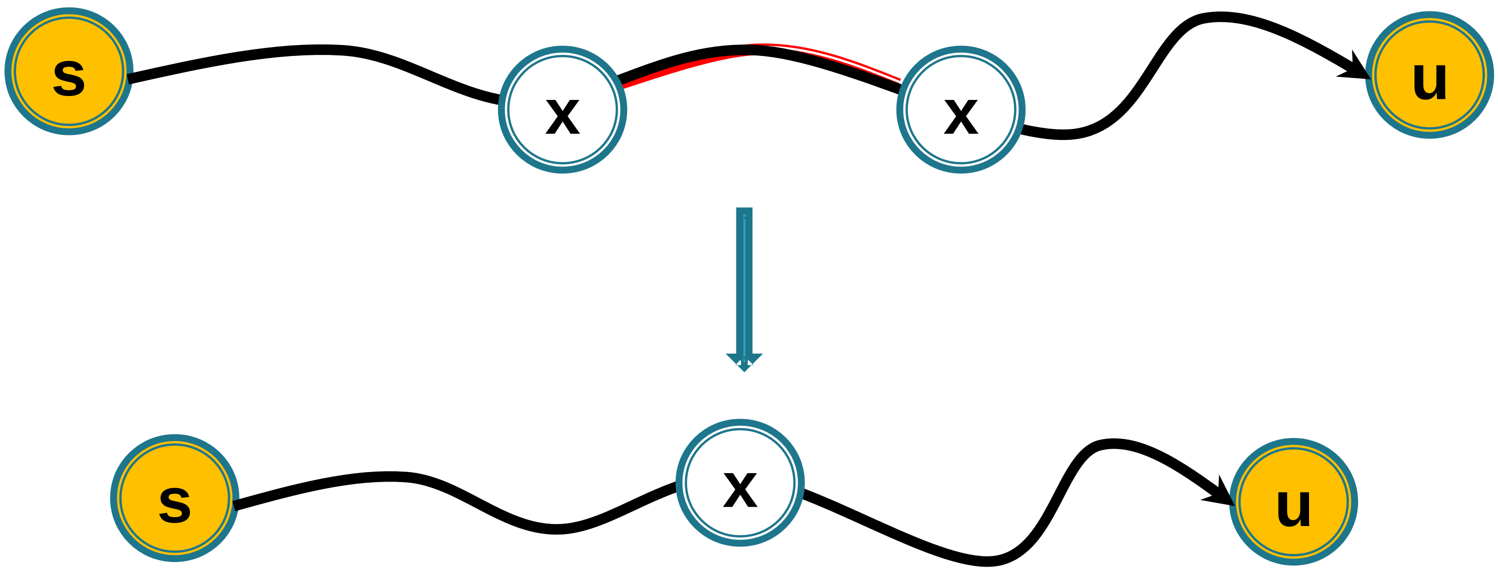
- între două vârfuri date
- de la un vârf la toate celelalte
- între oricare două vârfuri

Situaţii:

- Sunt acceptate şi arce de cost negativ?
- Graful conţine circuite?
- Graful conţine circuite de cost negativ?

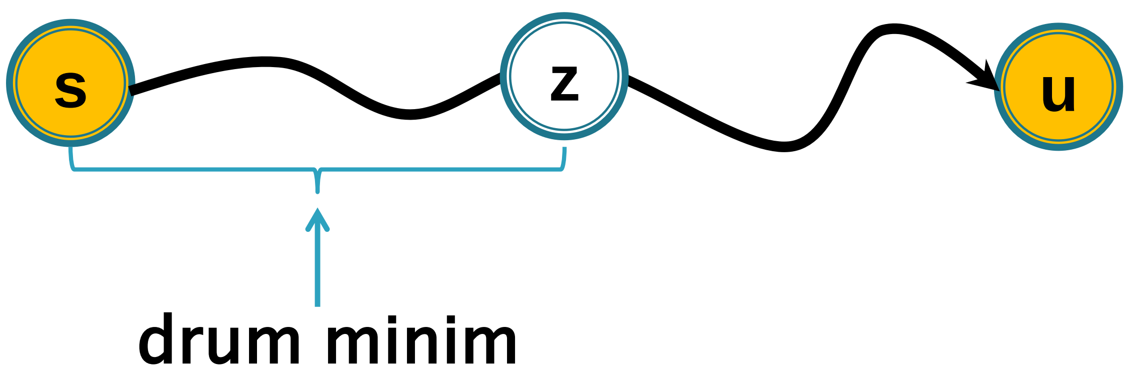
Observații

- **Observația 1.** Dacă P este un drum minim de la s la u și **nu există circuite de cost negativ**, atunci P este drum **elementar.**

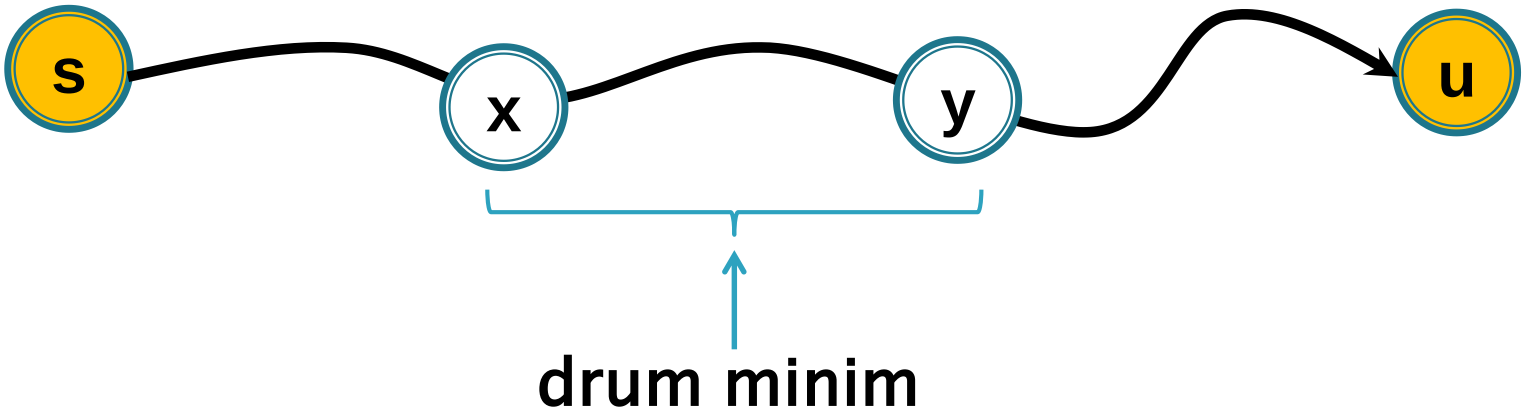


Observații

- **Observația 2.** Dacă P este un drum minim de la s la u și z este un vârf al lui P, atunci subdrumul lui P de la s la z este drum minim de la s la z.



Observații



Drumuri minime de la un vârf s dat la celelalte vârfuri (de sursă unică)

Problema drumurilor minime de sursă unică (de la s la celelalte vârfuri)

Se dau:

- un graf orientat ponderat G= (V, E, w), cu

  $w : E \rightarrow \mathbb{R}$

- un vârf de start s

Să se determine distanța de la s la fiecare vârf x al lui G / la un vârf dat t (și un arbore al distanțelor față de s/ un drum minim de la s la t)

Drumuri minime de sursă unică s

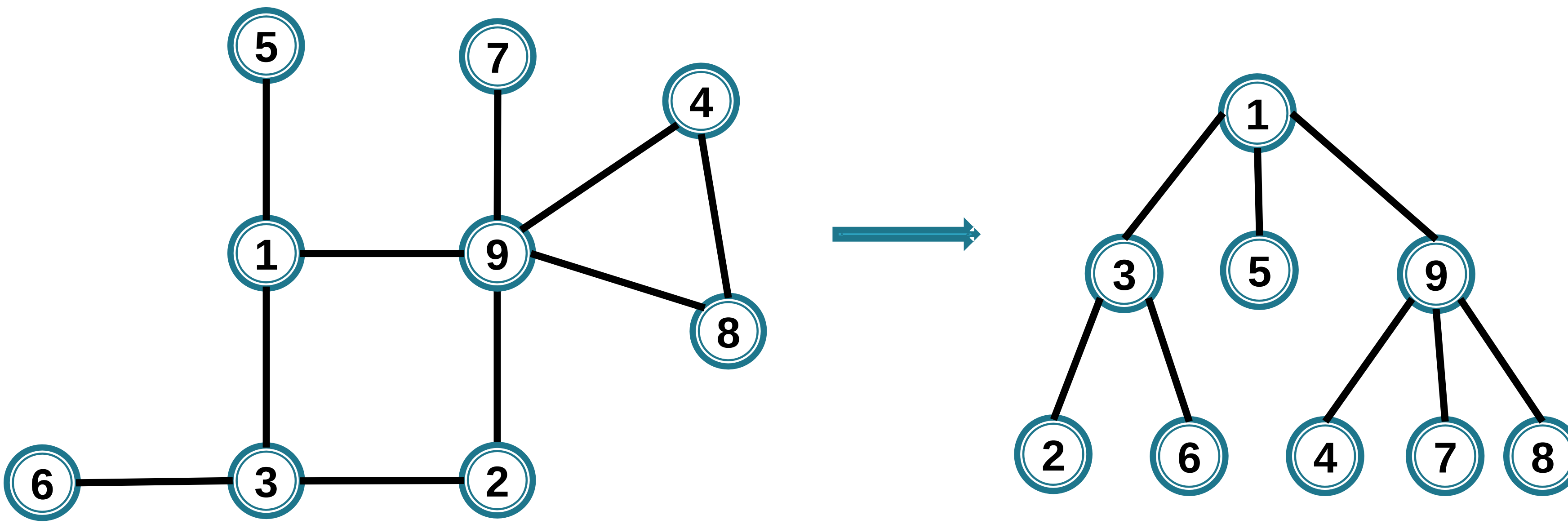
- Dacă G <u>nu</u> este ponderat, cum putem calcula distanţele față de s?

Drumuri minime de sursă unică s

- **Amintim**

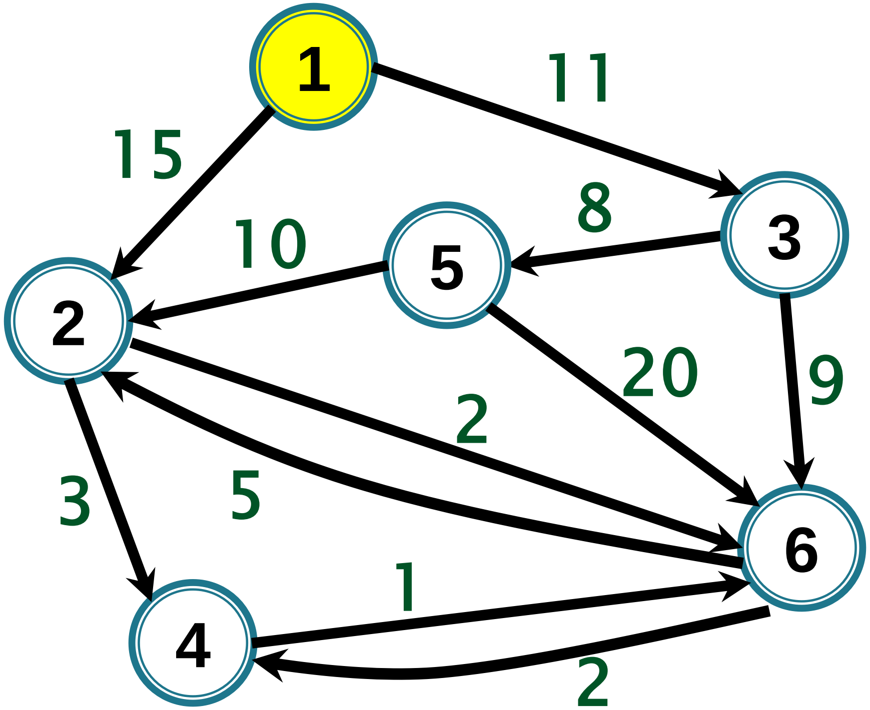
  În cazul unui graf neponderat, problema se rezolvă folosind parcurgerea BF din s

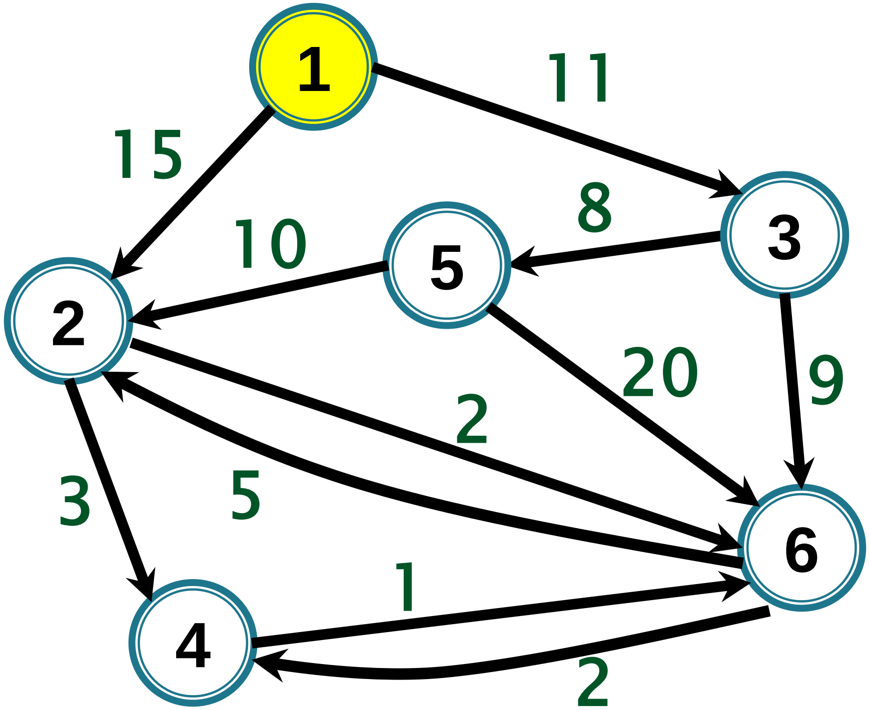
  $\Rightarrow$ arborele BF (al distanțelor față de s)

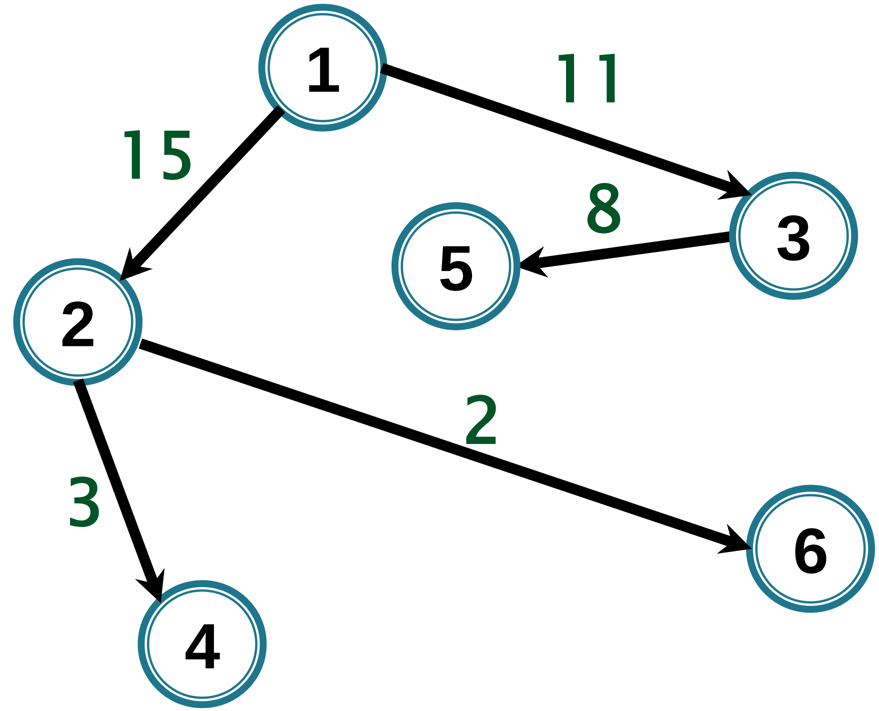


arborele BF

s = 1





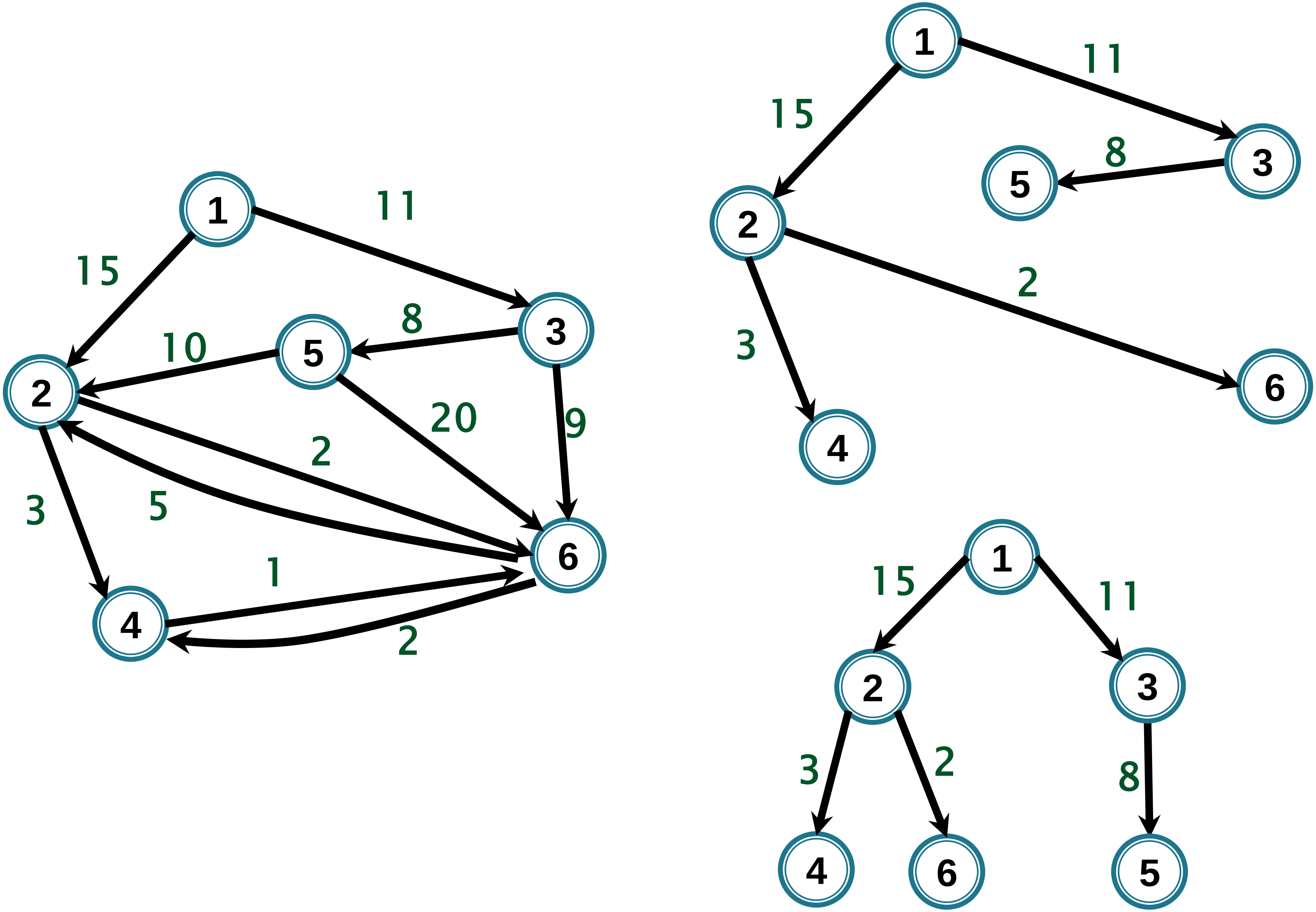


arbore al distanțelor față de 1

Definiție: **Pentru un vârf dat s** un <u>**arbore al distanțelor față de s**</u> = un subgraf T al lui G care **conservă distanțele** de la s la celelalte vârfuri accesibile din s

$$\delta_T(s, v) = \delta_G(s, v), \ \forall\, v \in V \text{ accesibil din } s,$$

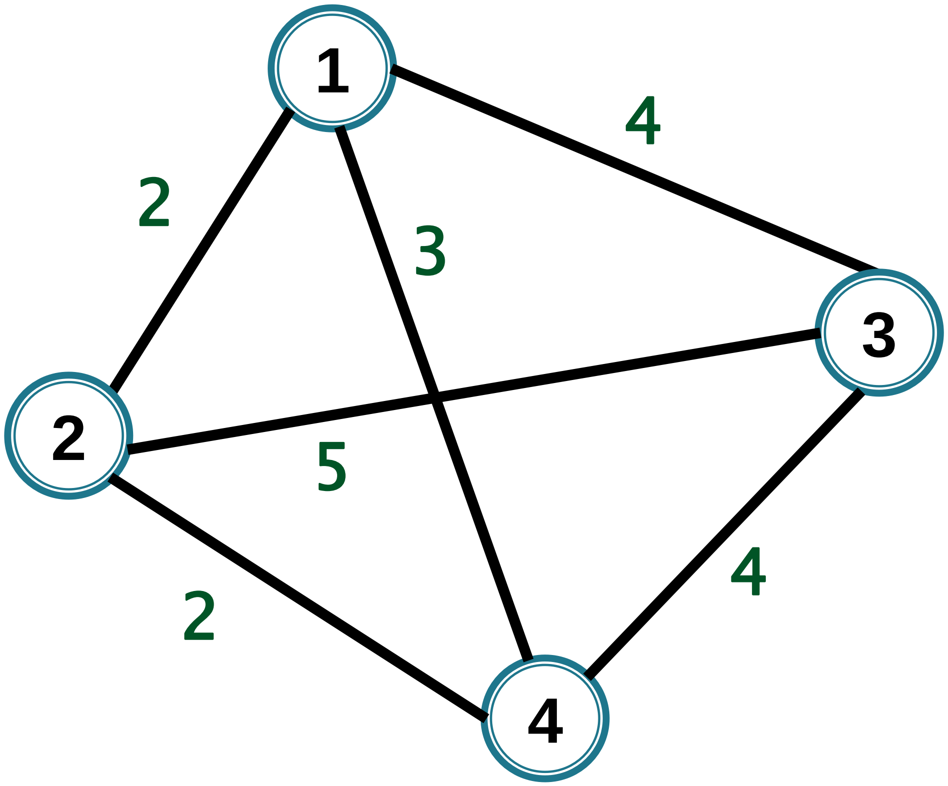
graful neorientat asociat lui T fiind arbore cu rădăcina în s (cu arcele corespunzătoare orientate de la s la frunze)



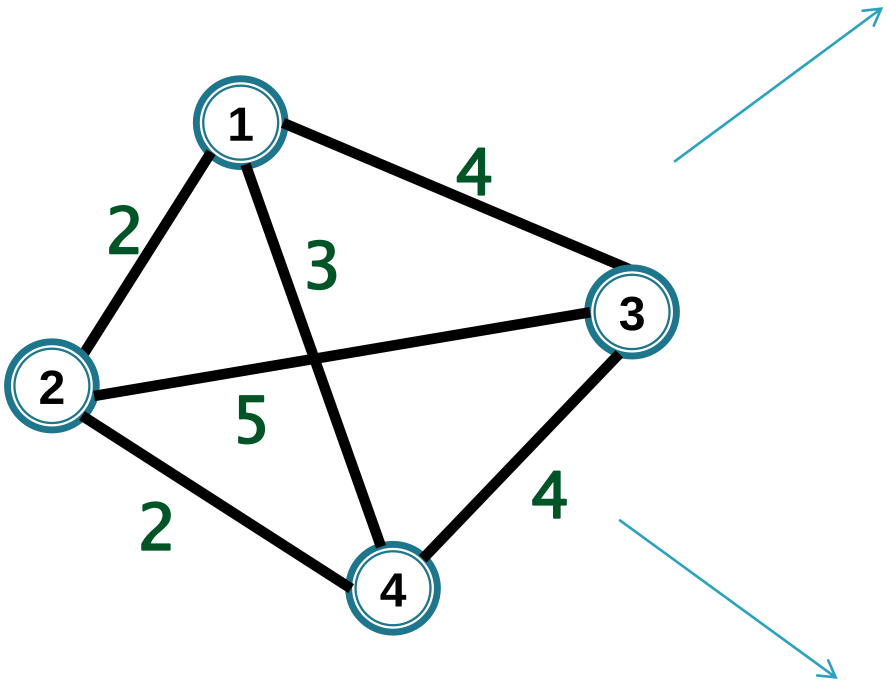
arbore al distanţelor faţă de 1

- Presupunem că toate vârfurile sunt accesibile din s
- Problema drumurilor minime de sursă unică este echivalentă cu determinarea unui arbore al distanțelor față de s

- Un arbore parțial de cost minim <u>nu</u> este neapărat un arbore de distanțe minime



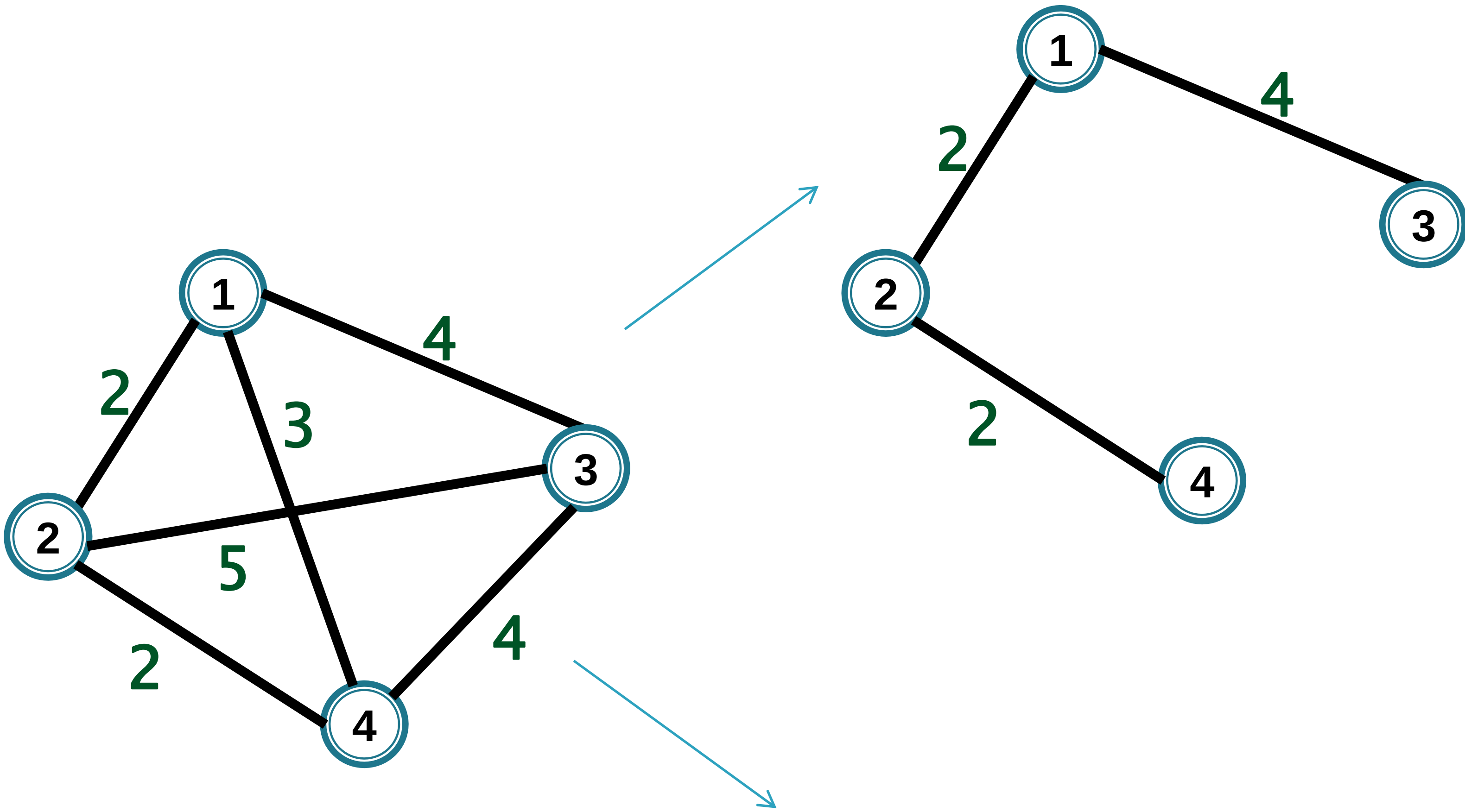
Un arbore parțial de cost minim nu este neapărat un arbore de distanțe minime



Apcm

Arbore al distanțelor față de 1

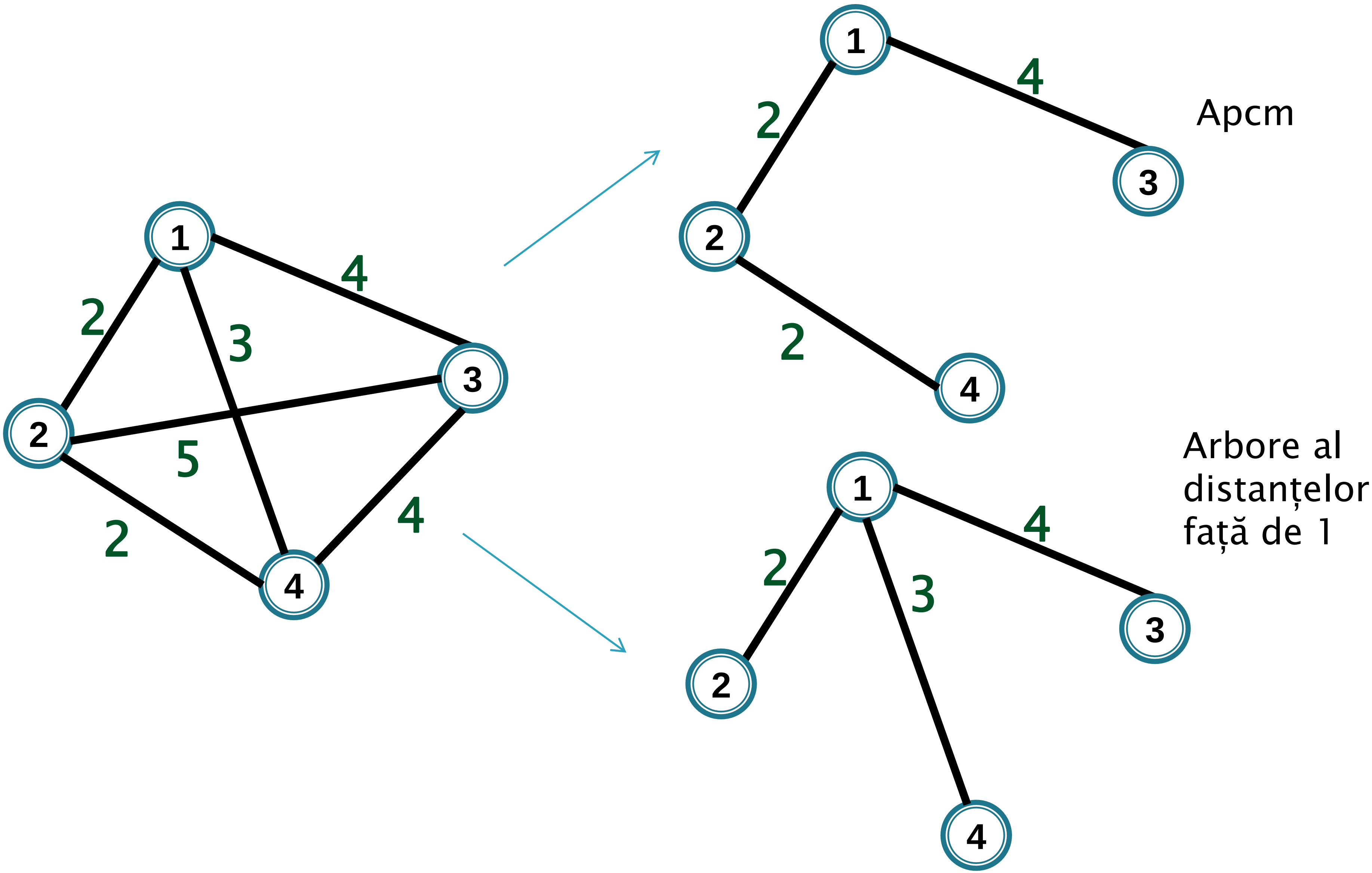
Un arbore parțial de cost minim nu este neapărat un arbore de distanțe minime



Apcm

Arbore al distanțelor față de 1

Un arbore parțial de cost minim nu este neapărat un arbore de distanțe minime

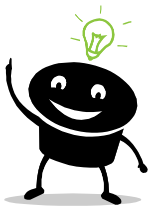


Drumuri minime de sursă unică s

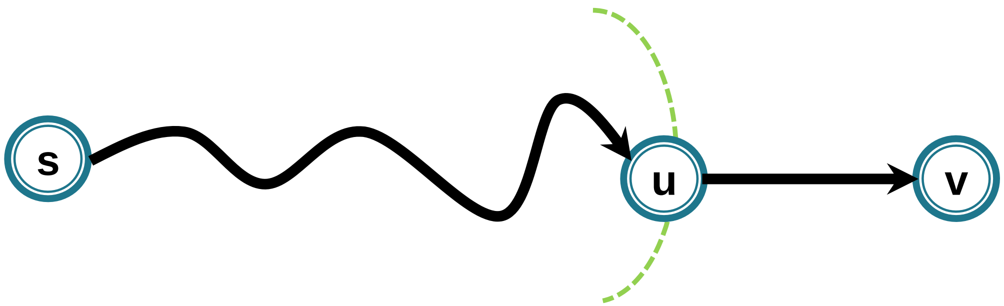
- În ce ordine considerăm vârfurile pentru a calcula distanţele faţă de s?

Drumuri minime de sursă unică s

- În ce ordine considerăm vârfurile pentru a calcula distanţele față de s?



"din aproape în aproape"



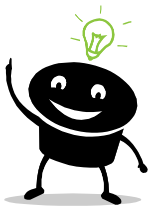
Dacă u este predecesor al lui v **pe un drum minim** de la s la v $\Rightarrow$

$$\delta(s,v) = \delta(s,u) + w(uv)$$

Ştim $\delta(s, u)$ $\Rightarrow$ aflăm și $\delta(s, v)$

Drumuri minime de sursă unică s

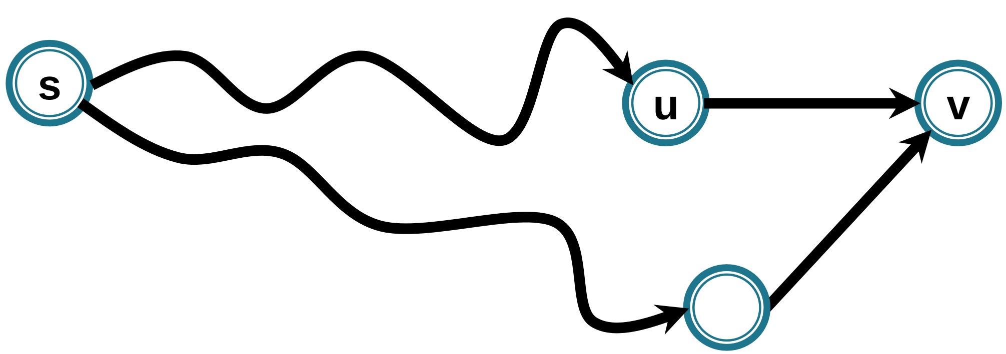
- În ce ordine considerăm vârfurile pentru a calcula distanţele faţă de s?



"din aproape în aproape" $\Rightarrow$

când considerăm un vârf v, pentru a calcula $\delta(s,v)$ ar fi util să ştim deja $\delta(s,u)$ pentru u cu $uv \in E$ (?!toate)

$$\delta(s,v) = \min\{ \delta(s,u) + w(u,v) \mid uv \in E \}$$



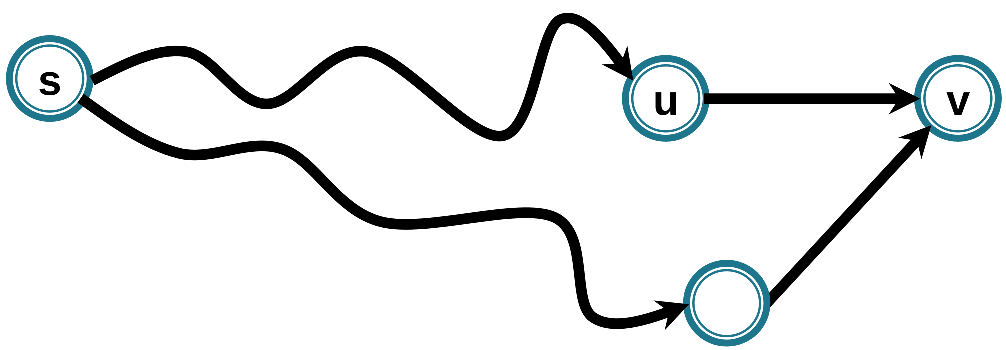
Drumuri minime de sursă unică s

- În ce ordine considerăm vârfurile pentru a calcula distanţele față de s?

  “din aproape în aproape” $\Rightarrow$

  când considerăm un vârf v, pentru a calcula $\delta(s,v)$ ar fi util să ştim deja $\delta(s,u)$ pentru orice u cu $uv \in E$

Ar fi utilă o ordonare a vârfurilor astfel încât dacă $uv \in E$, atunci u se află înaintea lui v



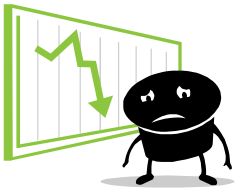
Drumuri minime de sursă unică s

- În ce ordine considerăm vârfurile pentru a calcula distanţele faţă de s?

  “din aproape în aproape” $\Rightarrow$

  când considerăm un vârf v, pentru a calcula $\delta(s,v)$ ar fi util să ştim deja $\delta(s,u)$ pentru orice u cu $uv \in E$

  - Ar fi utilă o ordonare a vârfurilor astfel încât dacă $uv \in E$, atunci u se află înaintea lui v



O astfel de ordonare <u>nu</u> <u>există</u> dacă graful conţine circuite

Drumuri minime de sursă unică s

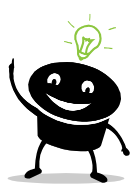
- În ce ordine considerăm vârfurile pentru a calcula distanţele față de s?

  "din aproape în aproape" $\Rightarrow$

  când considerăm un vârf v, pentru a calcula $\delta(s,v)$ ar fi util să știm deja $\delta(s,u)$ pentru orice u cu $uv \in E$

  - Ar fi utilă o ordonare a vârfurilor astfel încât dacă $uv \in E$, atunci u se află înaintea lui v

    O astfel de ordonare <u>nu</u> <u>există</u> dacă graful conţine circuite



Dacă există circuite – <u>estimăm</u> distanţele pe parcursul algoritmului şi considerăm vârful care este estimat a fi cel mai aproape de s

Drumuri minime de sursă unică s

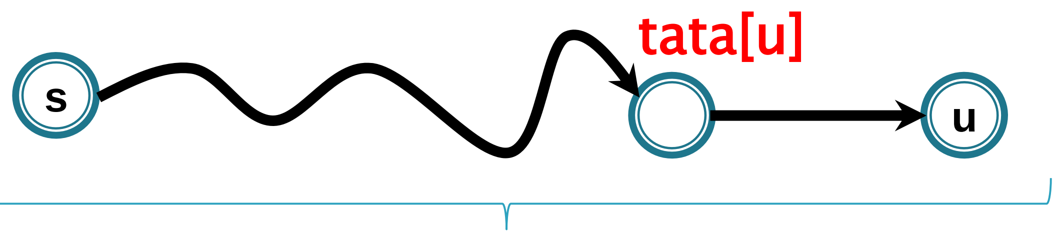
- Algoritmi pentru grafuri orientate cu circuite, dar cu ponderi pozitive – Dijkstra

- Algoritmi pentru grafuri orientate fără circuite (cu ponderi reale) DAGs = Directed Acyclic Graphs

- Algoritmi pentru grafuri orientate cu circuite și ponderi reale, care detectează existența de circuite negative – Bellman-Ford

Drumuri minime de sursă unică s

- **Idei comune:** **Pe parcursul algoritmului fiecare vârf are asociate informațiile:**
  - **d[u]** – etichetă de distanță
  - **tata[u]**

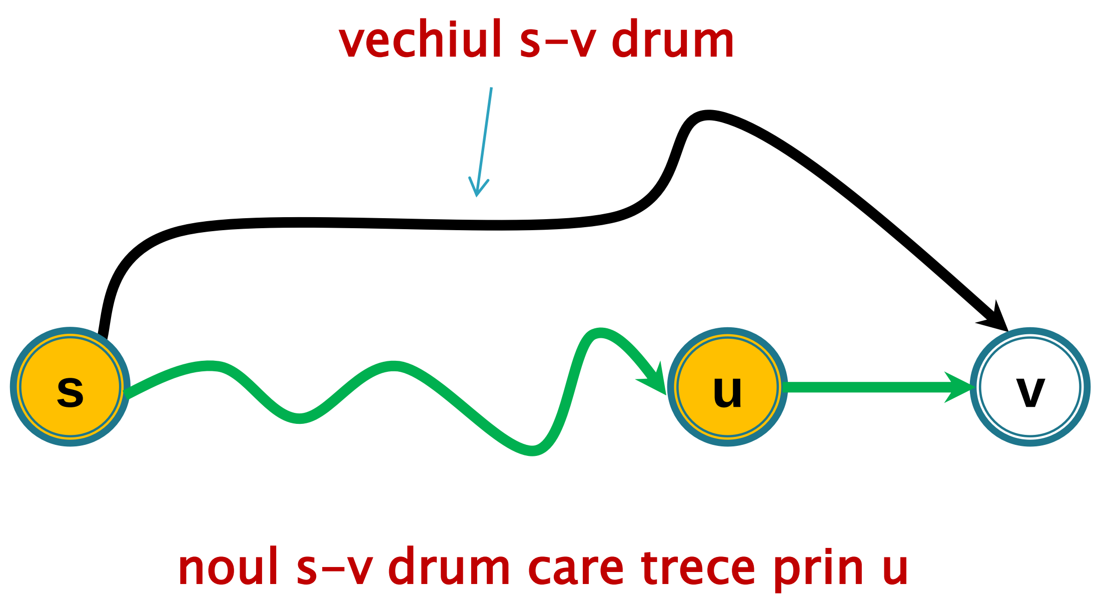


d[u] = costul minim al unui drum de la s la u descoperit până la acel moment

tata[u] = **predecesorul** lui u pe drumul de cost minim de la s la u descoperit până la acel moment

Drumuri minime de sursă unică s

- **Relaxarea unui arc (u, v)** = a verifica dacă d[v] poate fi îmbunătăţit trecând prin vârful u



Drumuri minime de sursă unică s

- Relaxarea unui arc (u, v) :

```
dacă d[u] + w(u,v) < d[v] atunci
      d[v] = d[u] + w(u,v);
      tata[v] = u
```