Second best

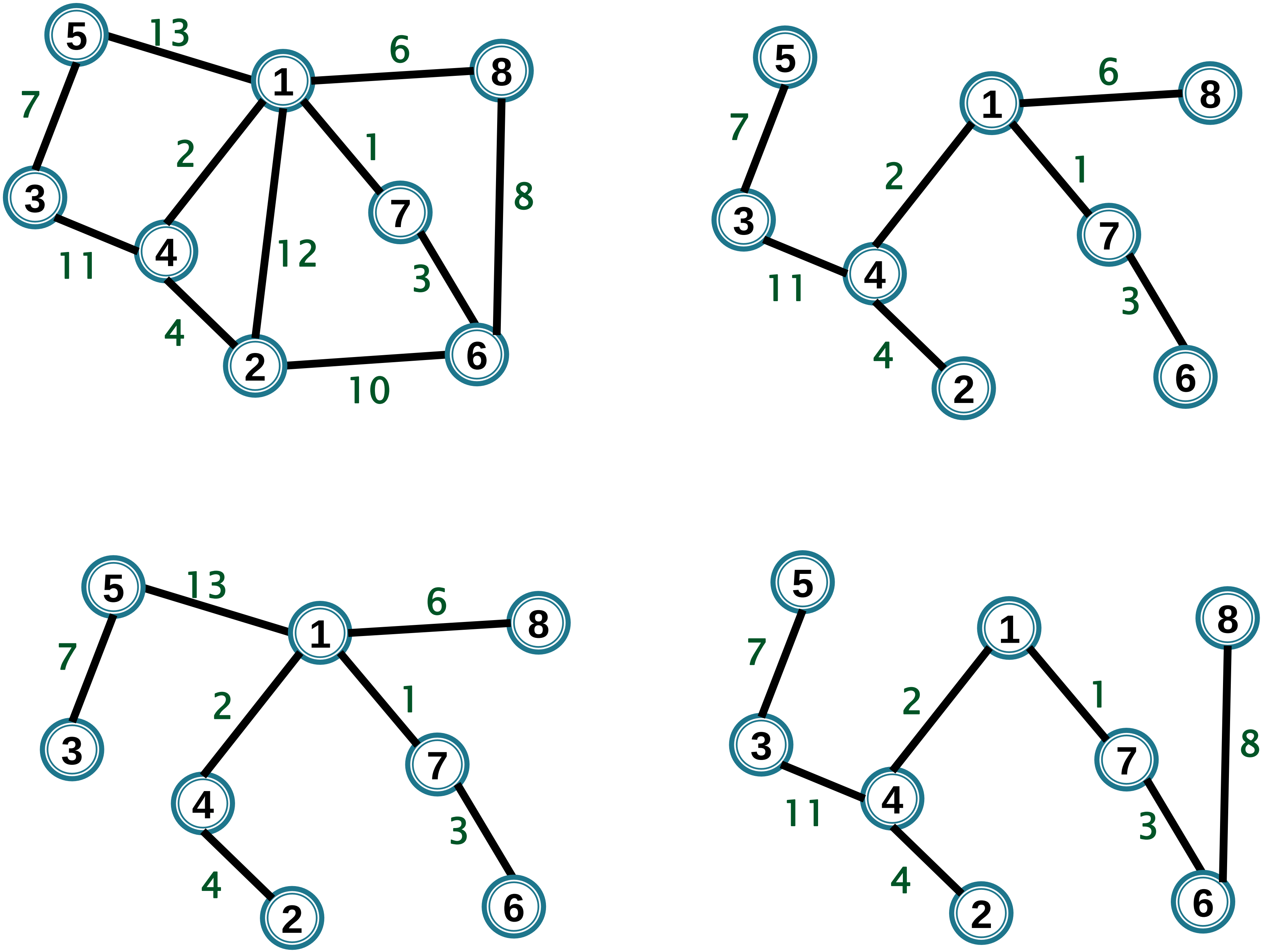
Fie G=(V,E,w) cu ponderi distincte

Atunci există un unic apcm T_{min} al lui G

Second best apcm – al doilea apcm = arbore parțial T_s cu

$$w(T_s) = \min\{w(T) \mid T \text{ arbore parțial în G diferit de } T_{min}\}$$

Second best – nu este neapărat unic



1
2
3
4
5
6
7
8
13
6
7
2
1
12
8
11
3
4
10

Second best

Cum se poate obtine second best din apcm?

Second best

Cum se poate obtine second best din apcm?

Idee:

second = apcm în care se schimbă doar o muchie

Second best

Propoziție Fie $G=(V,E,w)$ conex cu ponderi distincte

Fie T_{min} unicul apcm al lui G

Fie T_s un arbore second best

Atunci există $uv \in T_{min}$ și $xy \notin T_{min}$ astfel încât

$$T_s = T_{min} - uv + xy$$

Demonstrație - Tema

Second best

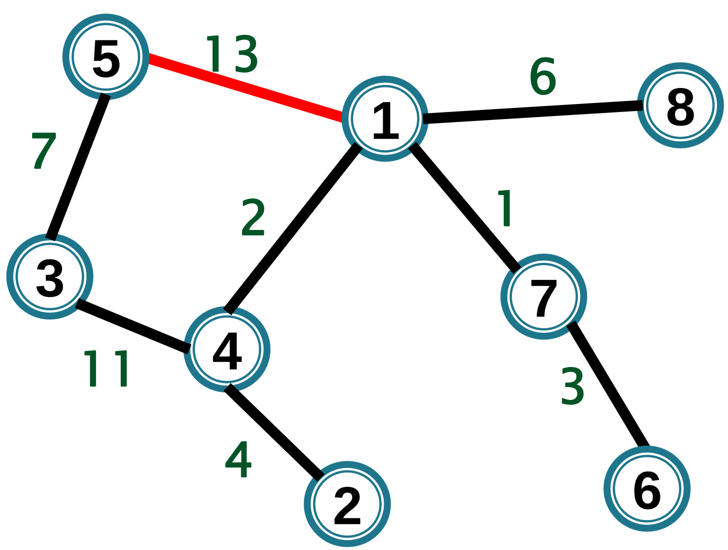
Idee algoritm:

Fie T_{min} apcm

Cum derminăm $uv \in T_{min}$ și $xy \notin T_{min}$ a.î

$T_{min} - uv + xy$ să fie arbore și

$w(xy) - w(uv)$ minim cu această proprietate?



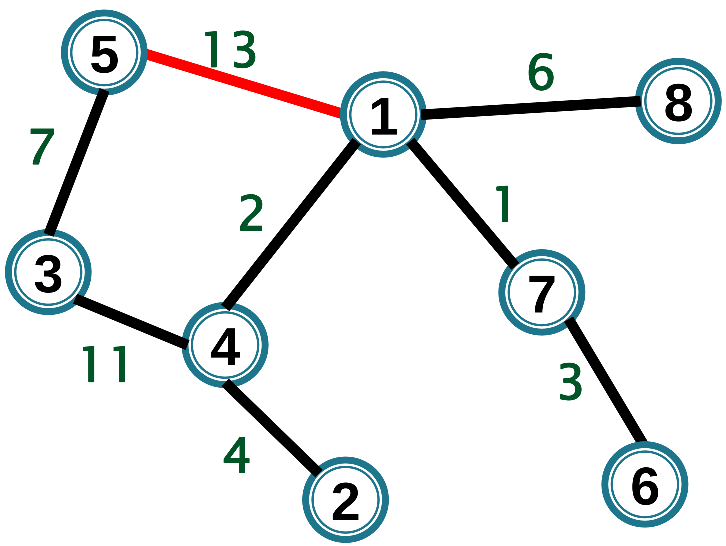
Second best

Idee algoritm:

Dacă **fixăm** un $xy \notin T_{min}$, atunci cum determinăm $uv \in T_{min}$ a.î.

T_{min} – uv +xy să fie arbore și

w(xy)–w(uv) minim cu această proprietate?



Second best

Idee algoritm:

Dacă **fixăm** un $xy \notin T_{min}$, atunci cum determinăm $uv \in T_{min}$ a.î

$T_{min} - uv + xy$ să fie arbore și

$w(xy) - w(uv)$ minim cu această proprietate?

- uv este muchia de cost maxim din ciclul închis de xy în T_{min}, adică muchia de cost maxim din lanțul de la x la y din T_{min}

Second best

Idee algoritm:

Fie T unicul apcm

Se determină xy pentru care se atinge minimul:

$$\min\{ w(xy) - w(\max[x,y]) \mid xy \notin T \}$$

unde

max[x,y] = **muchia** maximă din lanțul de la x la y din T

$T_s = T + xy - \max[x,y]$

Second best

Algoritm second best

1. Determinăm T apcm în G

2. Pentru orice $x, y \in T$ determină:

   $max[x,y]$ = **muchia** maximă din lanțul de la x la y din T

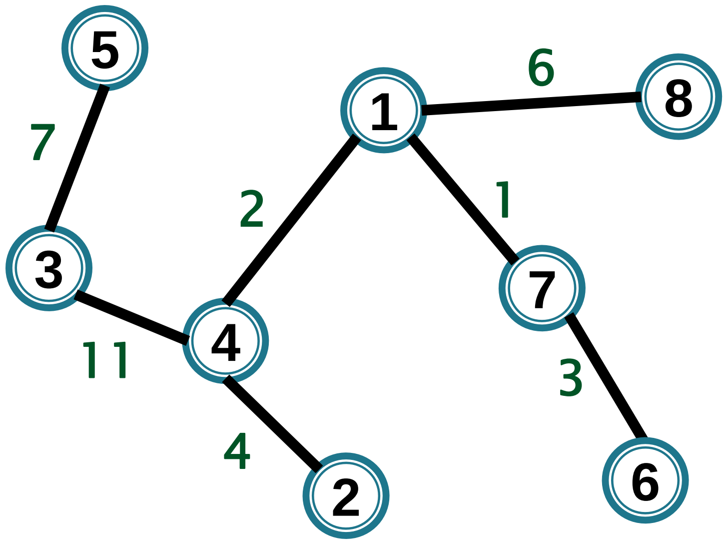
3. Determină o muchie $xy \notin T$ cu

   $w(x,y) - w(max[x,y])$ minim

4. $T_s = T + xy - max[x,y]$

Second best

Cum determinăm max[s,x] pentru orice s,x in T ?

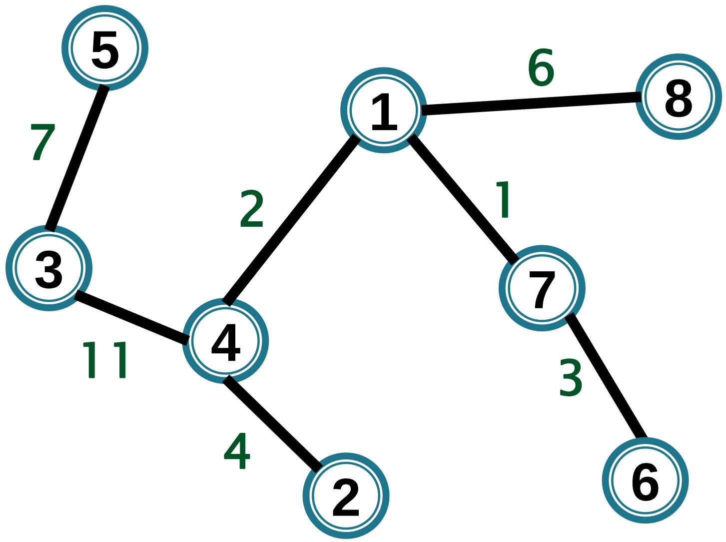


arbore parțial de cost minim

x	1	2	3	4	5	6	7	8
max[1,x]								
max[2,x]								

Second best

Cum determinăm max[s,x] pentru orice s,x in T ?



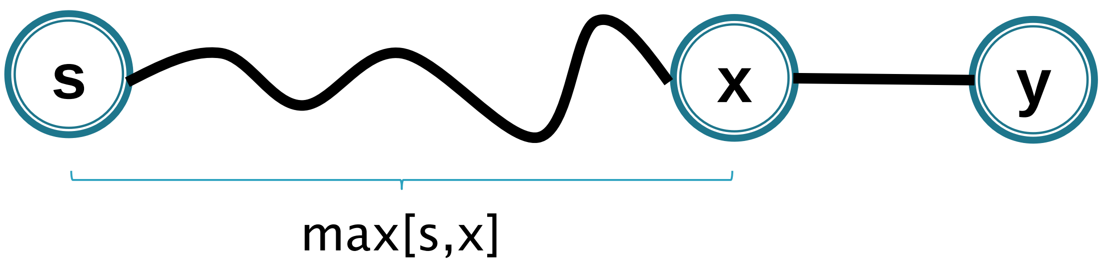
arbore parțial de cost minim

x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2) 4	(4,3) 11	(1,4) 2	(4,3) 11	(7,6) 3	(1,7) 1	(1,8) 6
max[2,x]	(2,4) 2	0	(3,4) 11	(2,4) 4	(3,4) 11	(2,4) 4	(2,4) 4	(1,8) 6

Second best

Determinare max[s,x] pentru orice s,x in T

Pentru s fixat - **parcurgere din s**

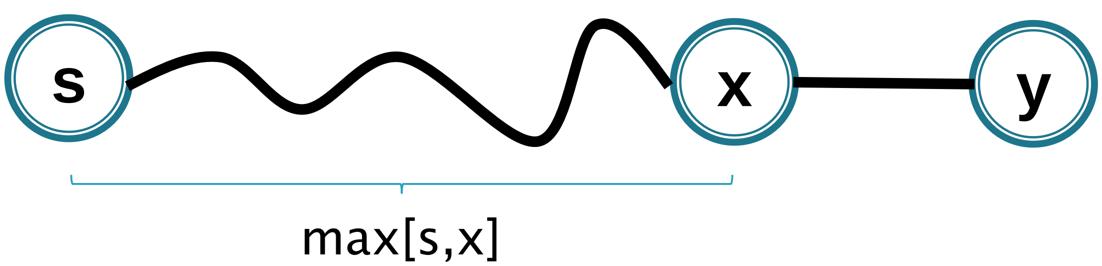


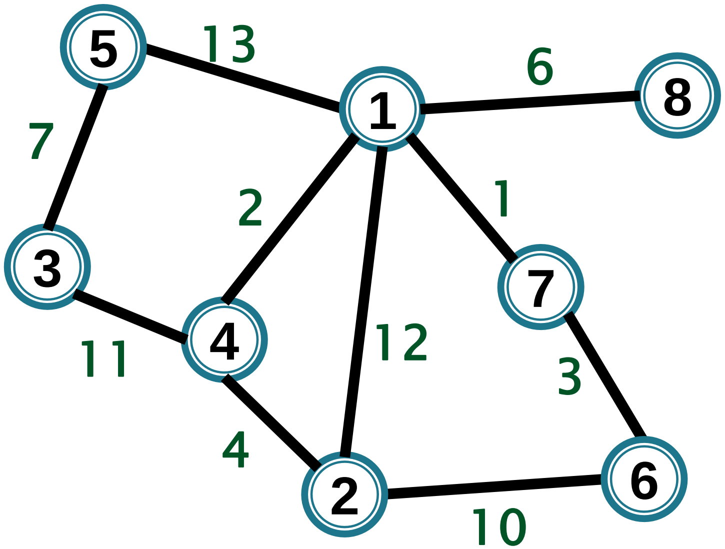
Second best

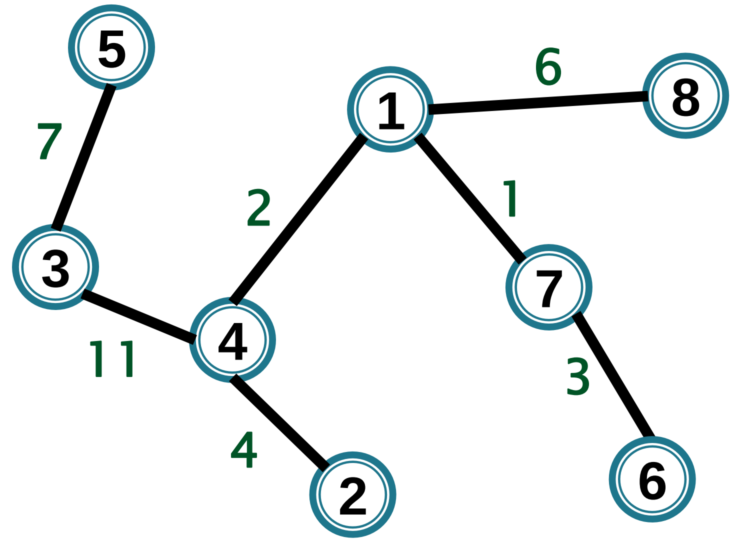
Determinare max[s,x] pentru orice s,x în T

Pentru s fixat – **parcurgere din s**, actualizând pentru un vârf y descoperit din x max[s,y] astfel:

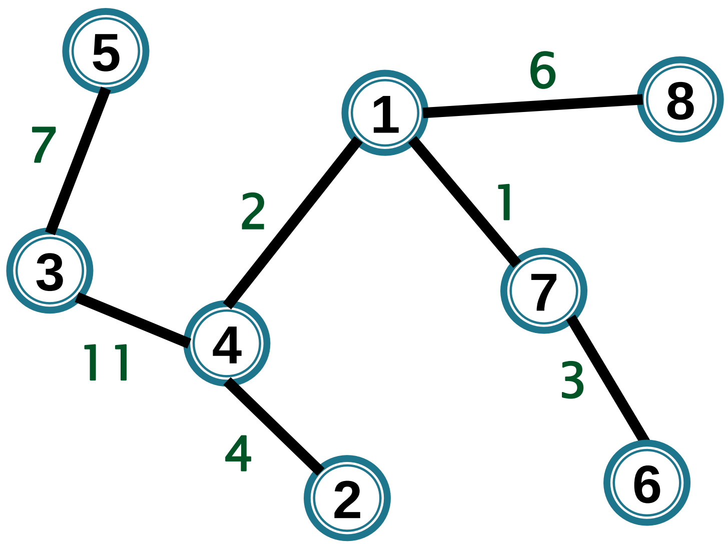
$$\text{max[s,y]} = \begin{cases} xy, & \text{dacă } w(x,y) > w(\text{max[s,x]}) \\ \text{max[s,x]}, & \text{altfel} \end{cases}$$





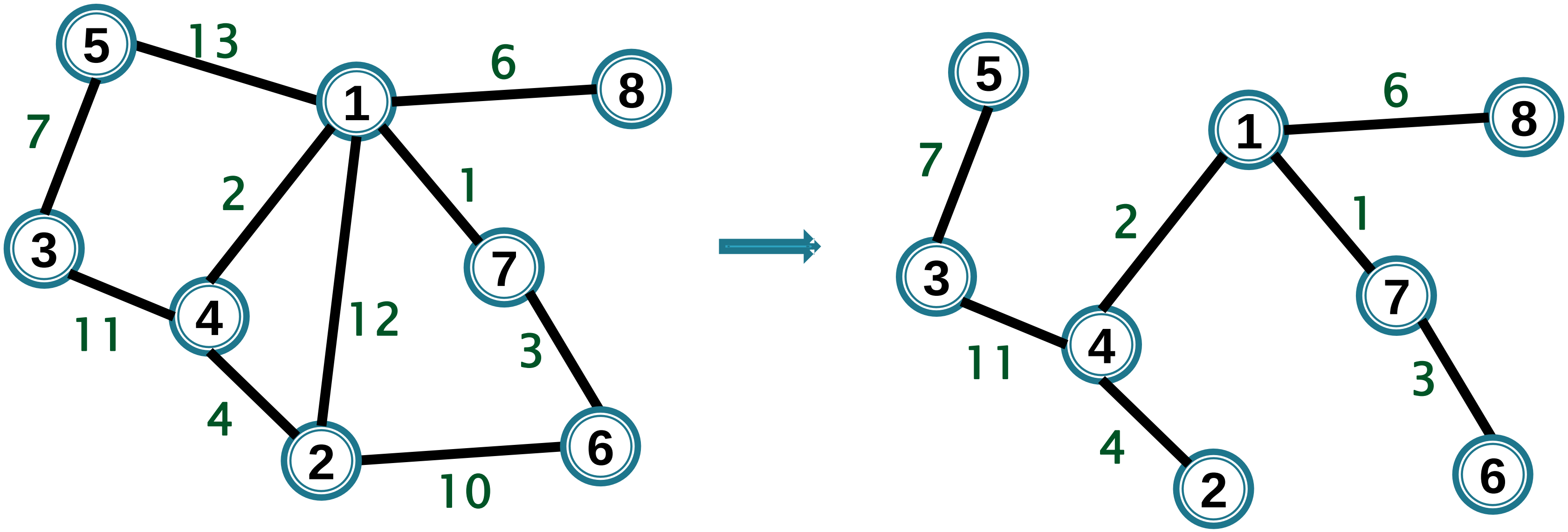


arbore parțial de cost minim

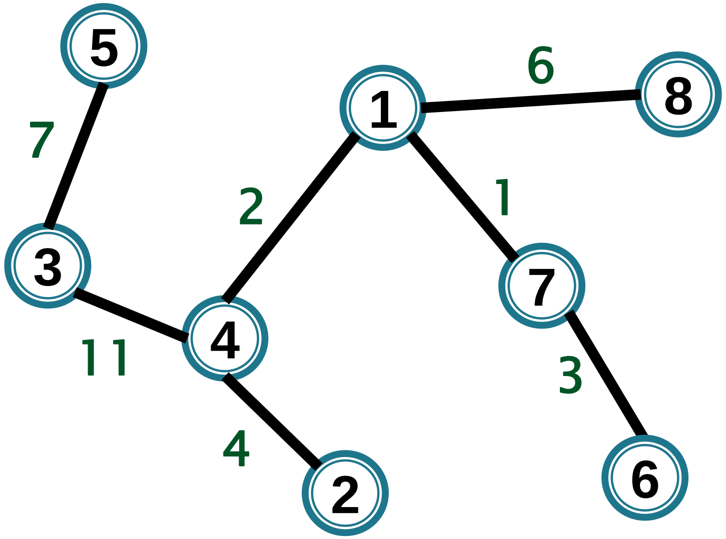


arbore parțial de cost minim

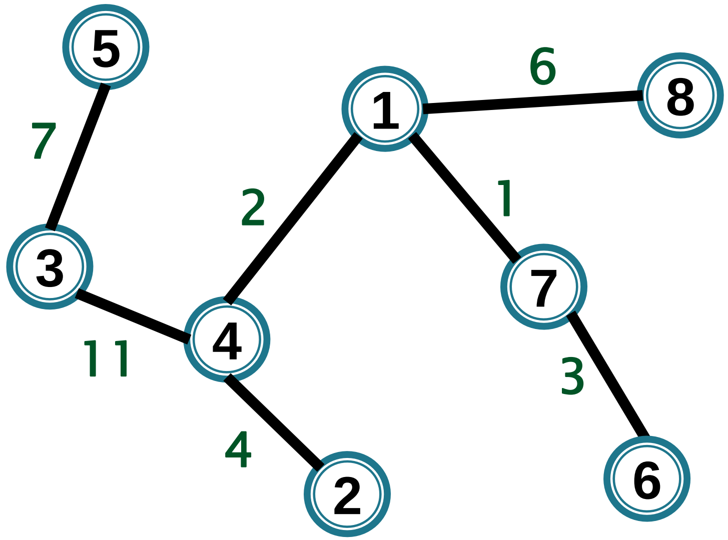
Calculăm max[1, x] folosind BF(1)

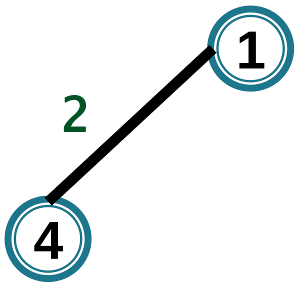


arbore parțial de cost minim



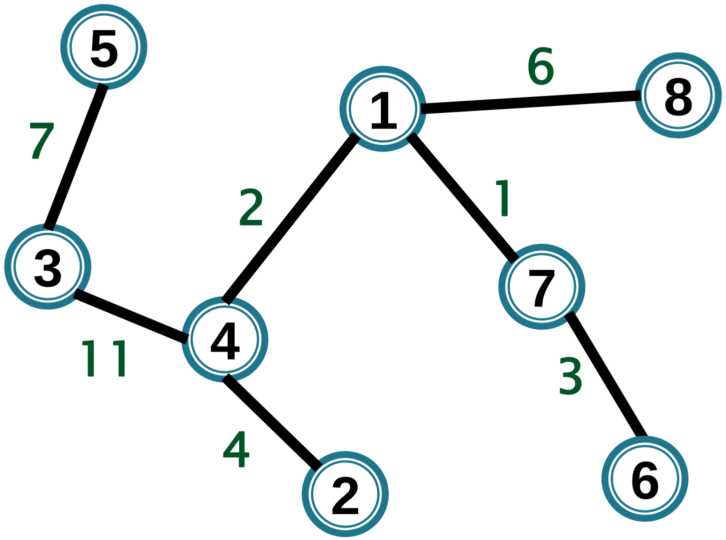
x	1	2	3	4	5	6	7	8
max[1,x] d	0							

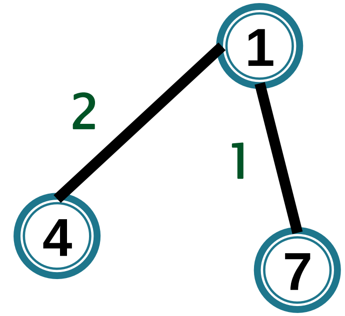




`max[1,4] = max{ w(1,4) , max[1,1] } => (1,4)`

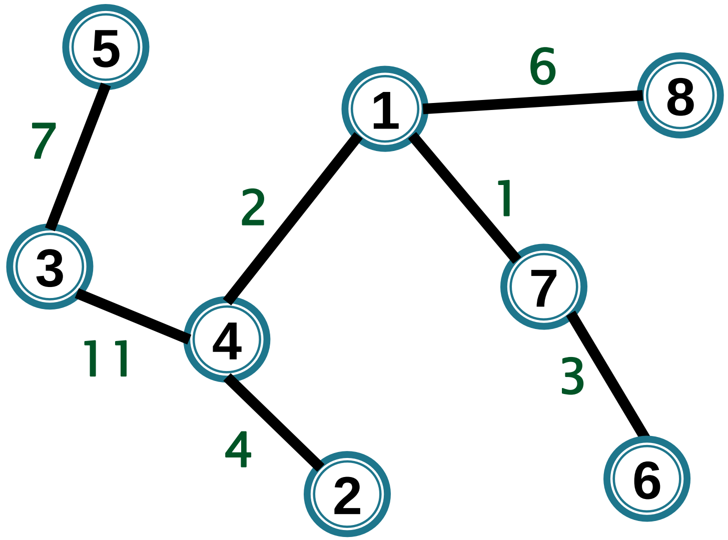
x	1	2	3	4	5	6	7	8
max[1,x] d	0			(1,4) 2				

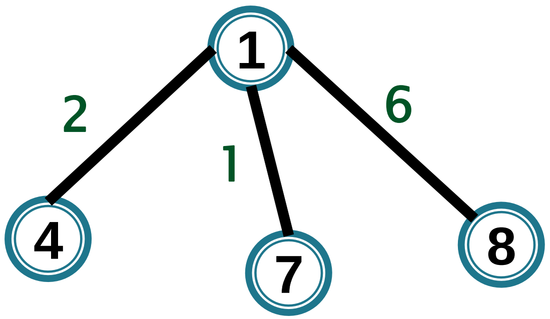





```
max[1,7] = max{ w(1,7) , max[1,1] } => (1,7)
```

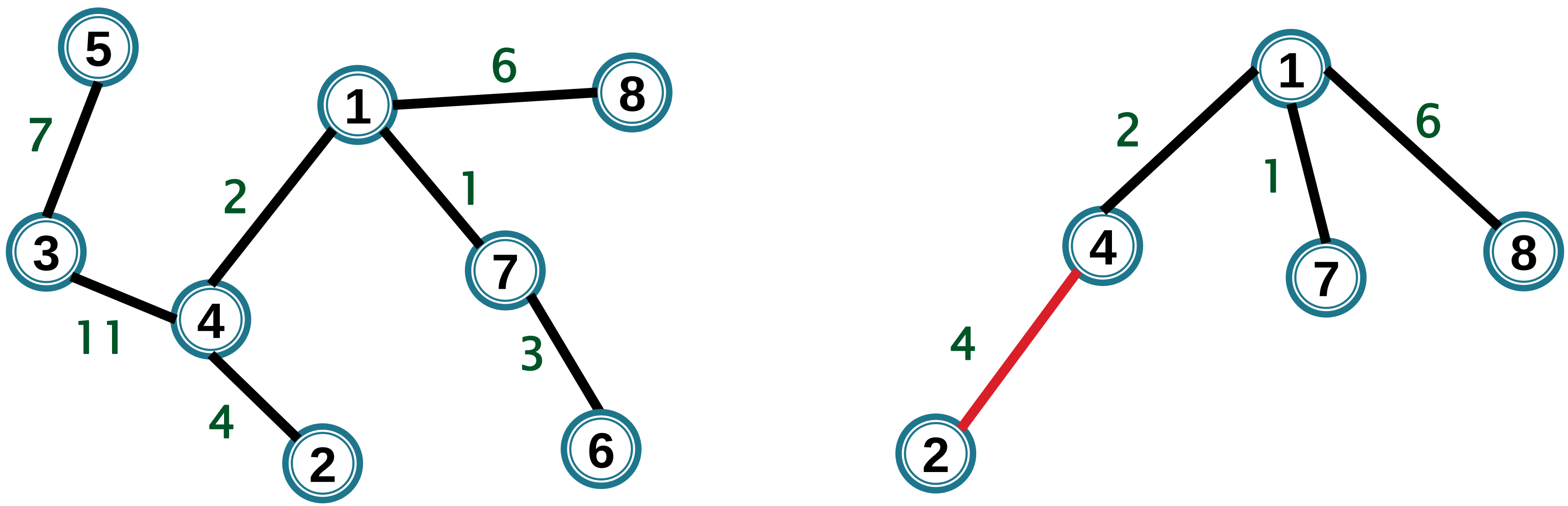
x	1	2	3	4	5	6	7	8
max[1,x]	0			(1,4) 2			**(1,7)** **1**	





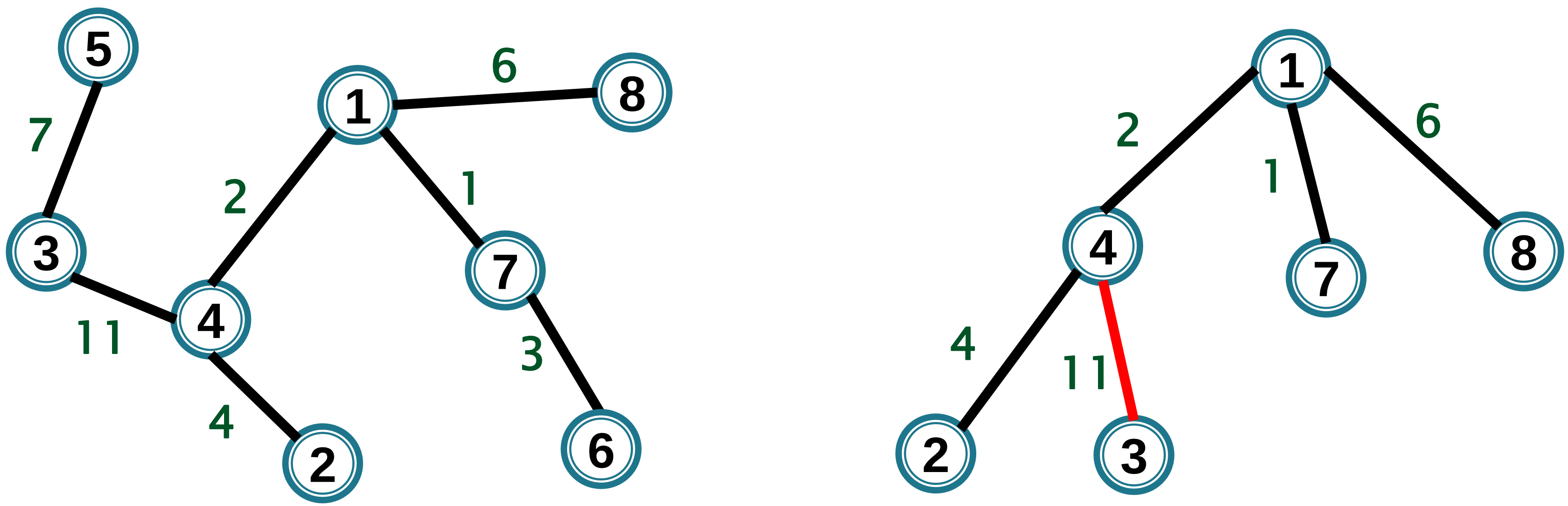
max[1,8] = max{ w(1,8) , max[1,1] } => (1,8)

x	1	2	3	4	5	6	7	8
max[1,x]	0			(1,4) 2			(1,7) 1	**(1,8) 6**



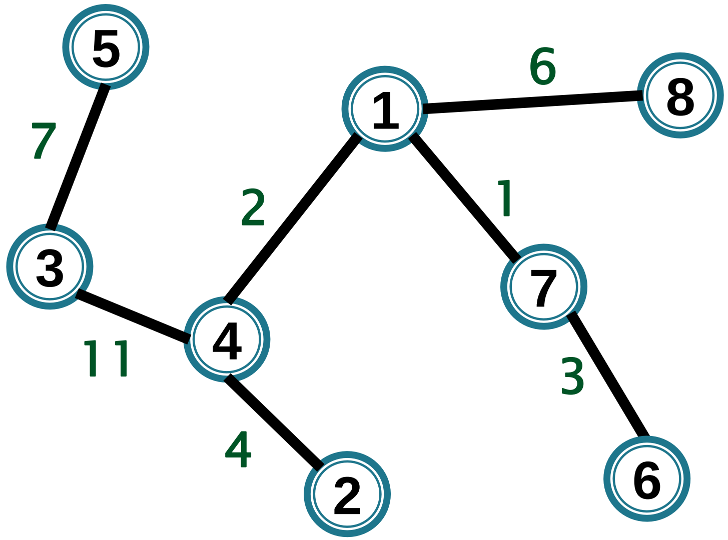
max[1,2] = max{ w(4,2) , max[1,4] } => (4,2)

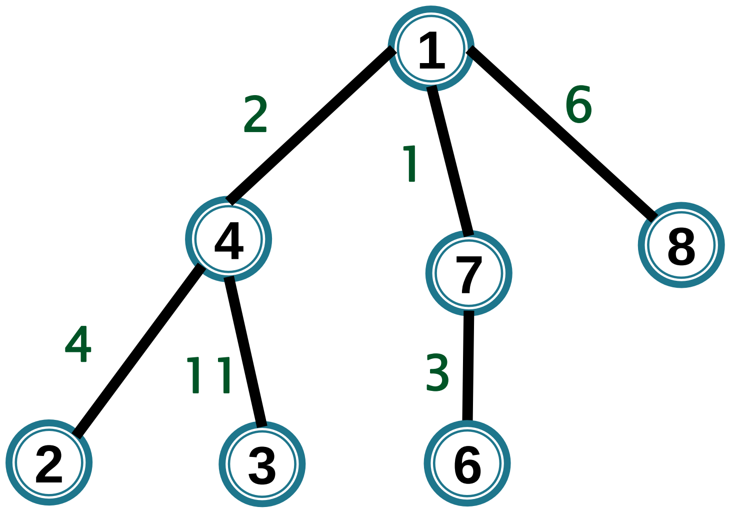
x	1	2	3	4	5	6	7	8
max[1,x]	0	**(4,2)** **4**		(1,4) 2			(1,7) 1	(1,8) 6



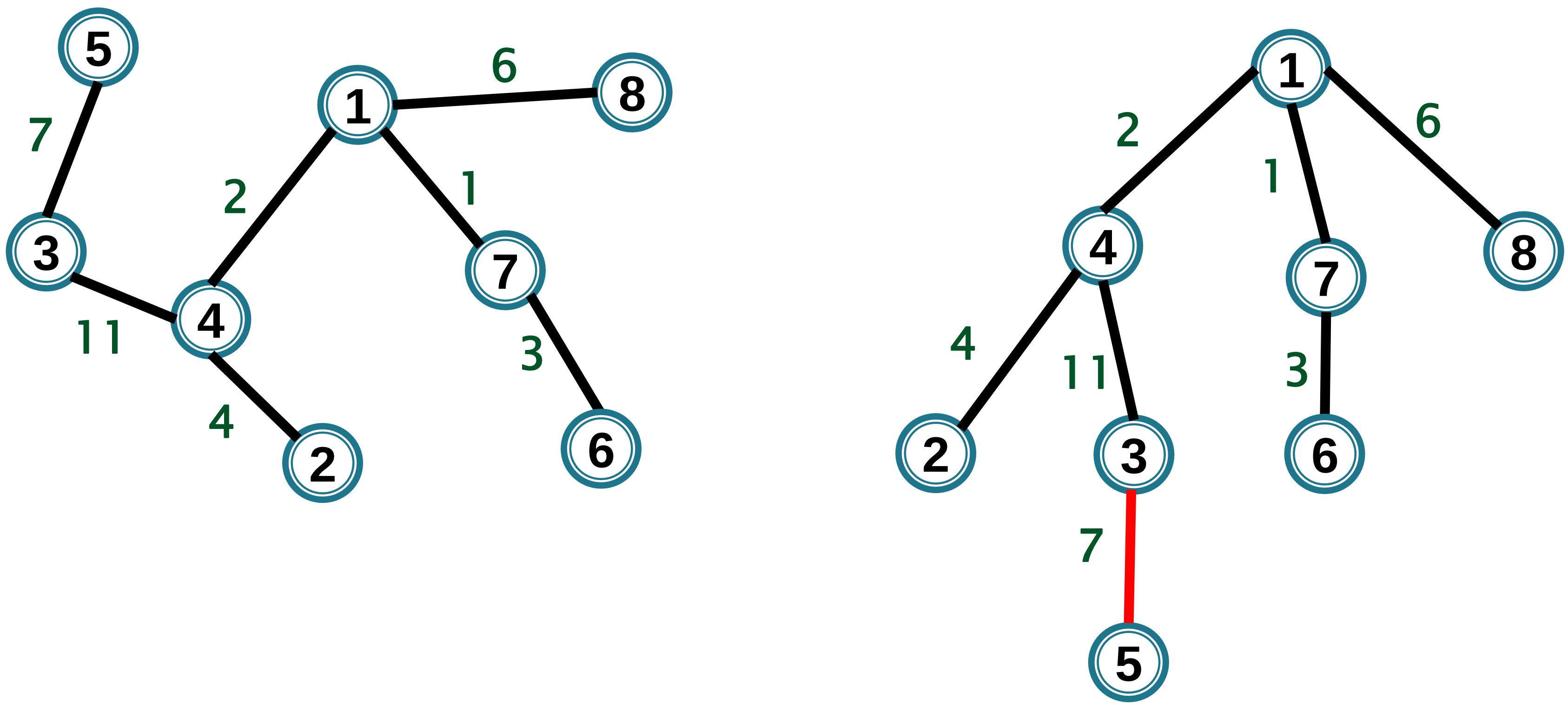
max[1,3] = max{ w(4,3) , max[1,4] } => (4,3)

x	1	2	3	4	5	6	7	8
max[1,x] d	0	(4,2) 4	**(4,3)** **11**	(1,4) 2			(1,7) 1	(1,8) 6



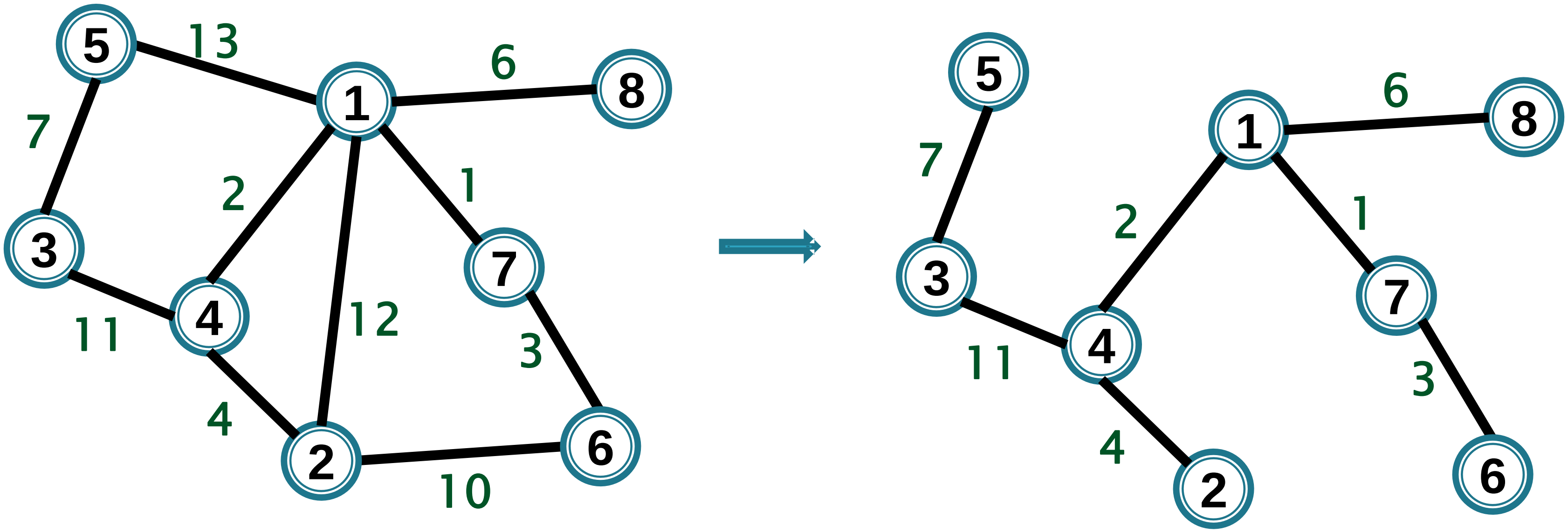


x	1	2	3	4	5	6	7	8
max[1,x] d	0	(4,2) 4	(4,3) 11	(1,4) 2		**(7,6)** **3**	(1,7) 1	(1,8) 6



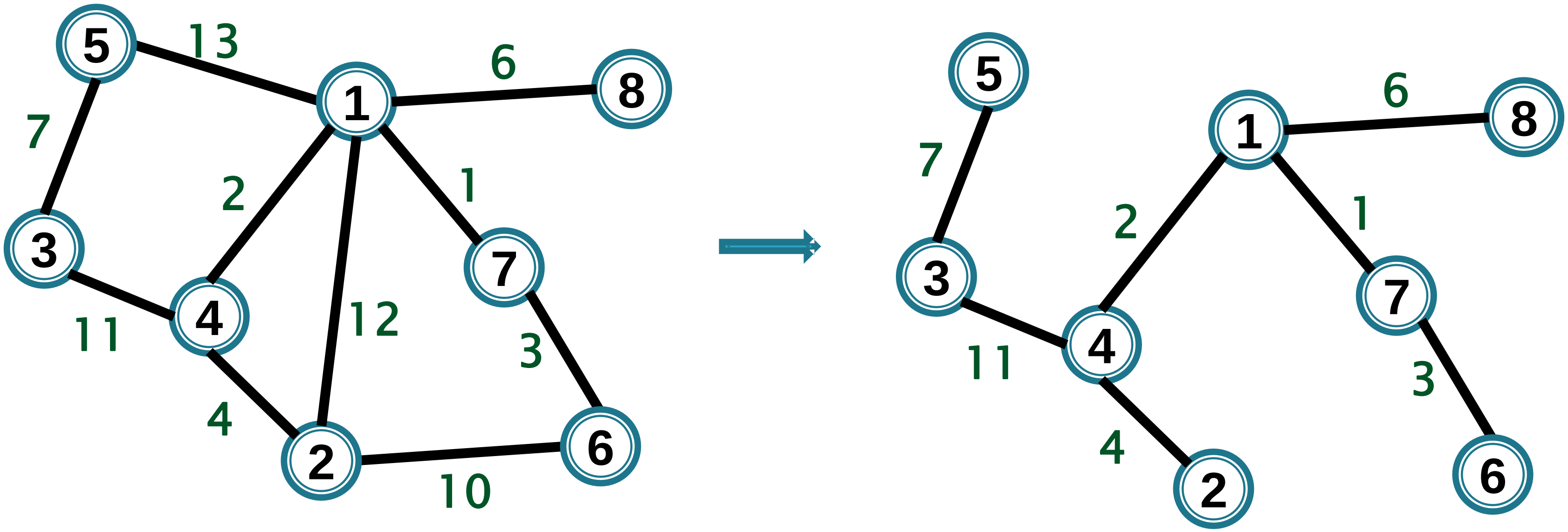
max[1,5] = max{ w(3,5) , max[1,3] } = max[1,3] => (4,3)

x	1	2	3	4	5	6	7	8
max[1,x] d	0	(4,2) 4	(4,3) 11	(1,4) 2	**(4,3) 11**	(7,6) 3	(1,7) 1	(1,8) 6

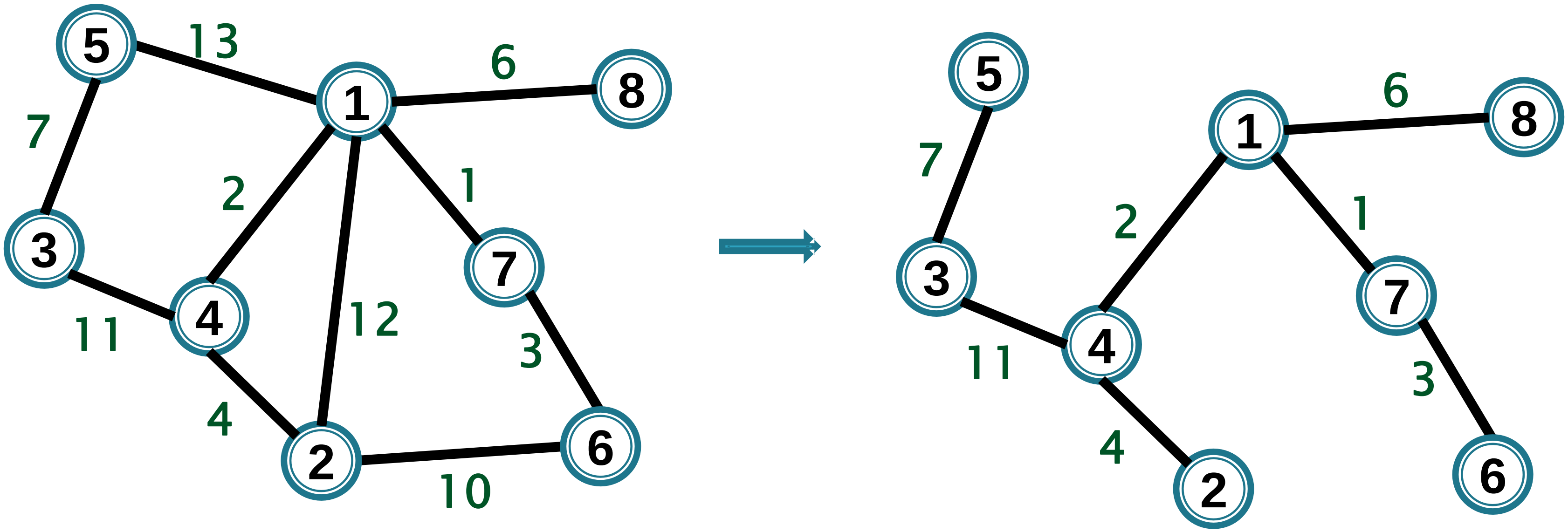


arbore parțial de cost minim

Calculăm max[2, x] folosind BF(2)

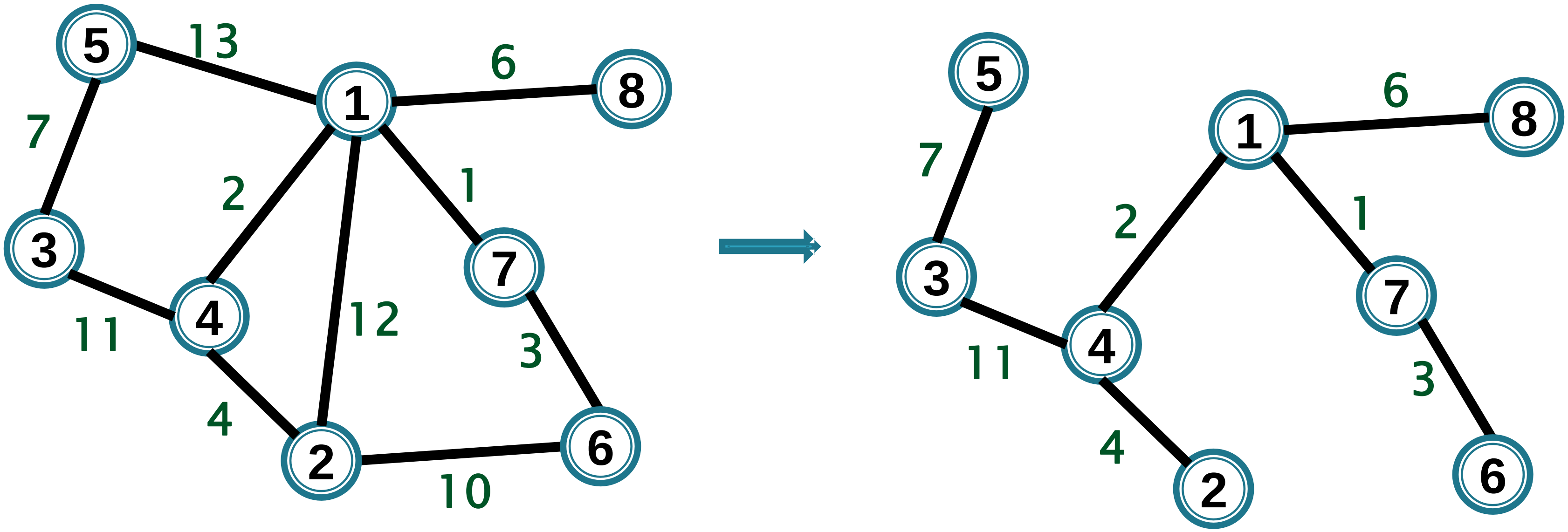


x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)



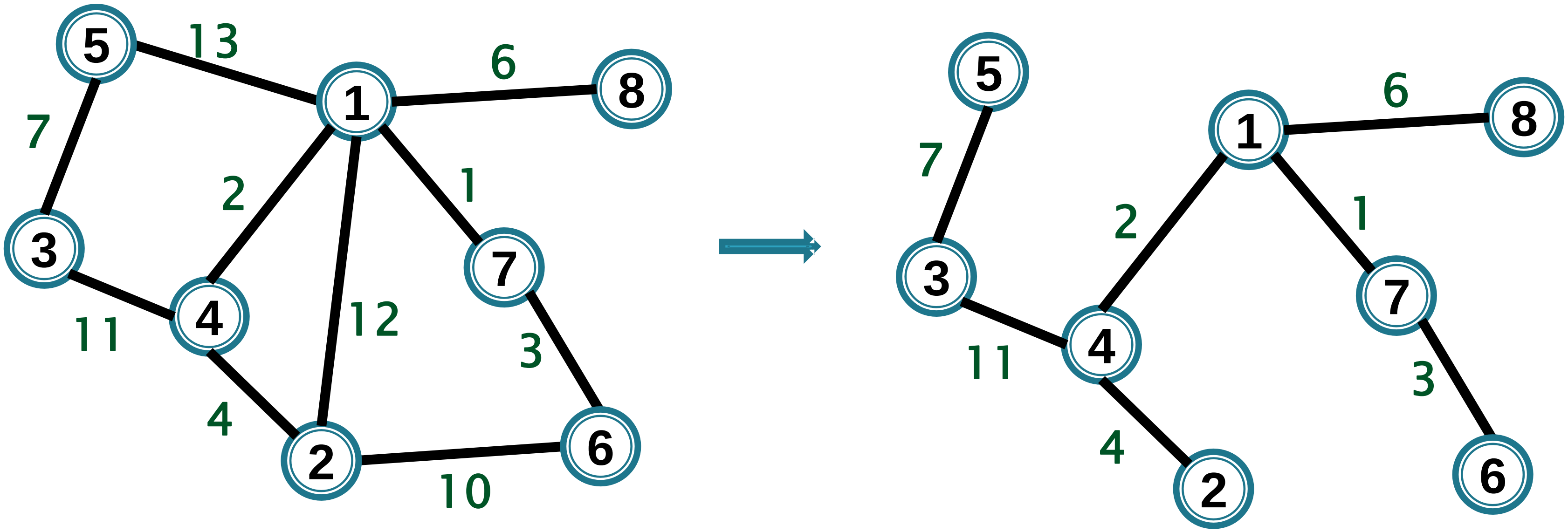
x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

Considerăm pe rând muchiile care nu sunt în apcm:



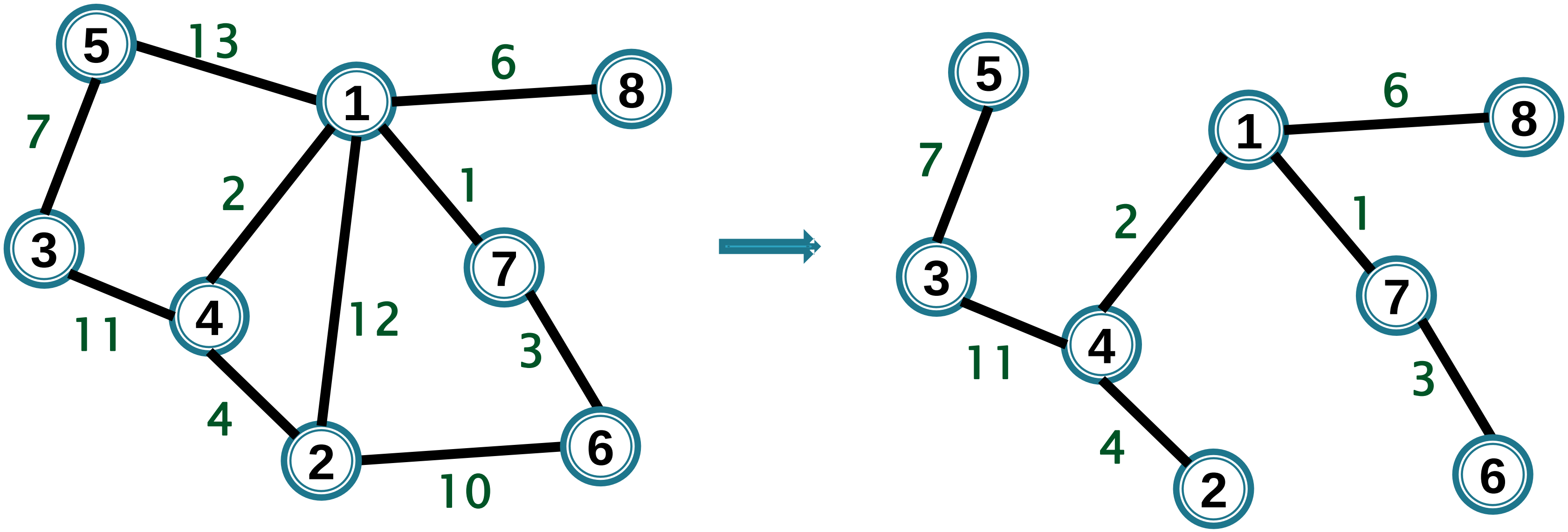
x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

(2,6):



x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

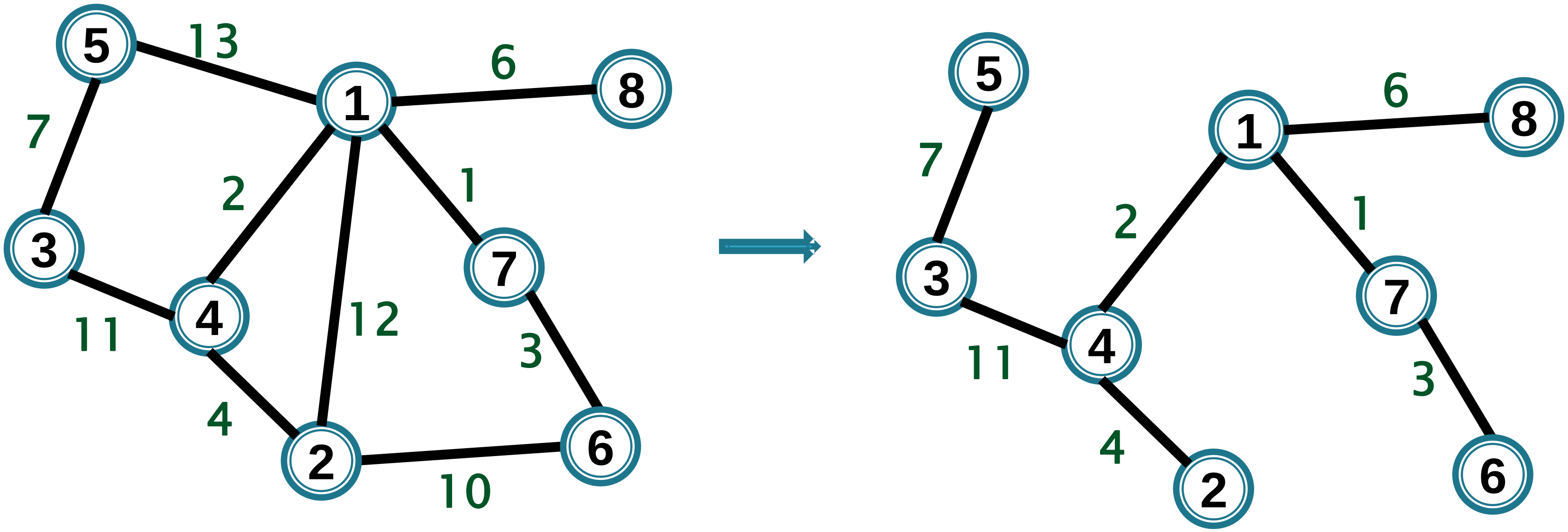
(2,6): w(2,6) – w(max[2,6]) = w(2,6) – w(2,4) = 10 – 4 = 6



x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

(2,6): $w(2,6) - w(max[2,6]) = w(2,6) - w(2,4) = 10 - 4 = 6$

(1,2): $w(1,2) - w(max[1,2]) = w(1,2) - w(2,4) = 12 - 4 = 8$

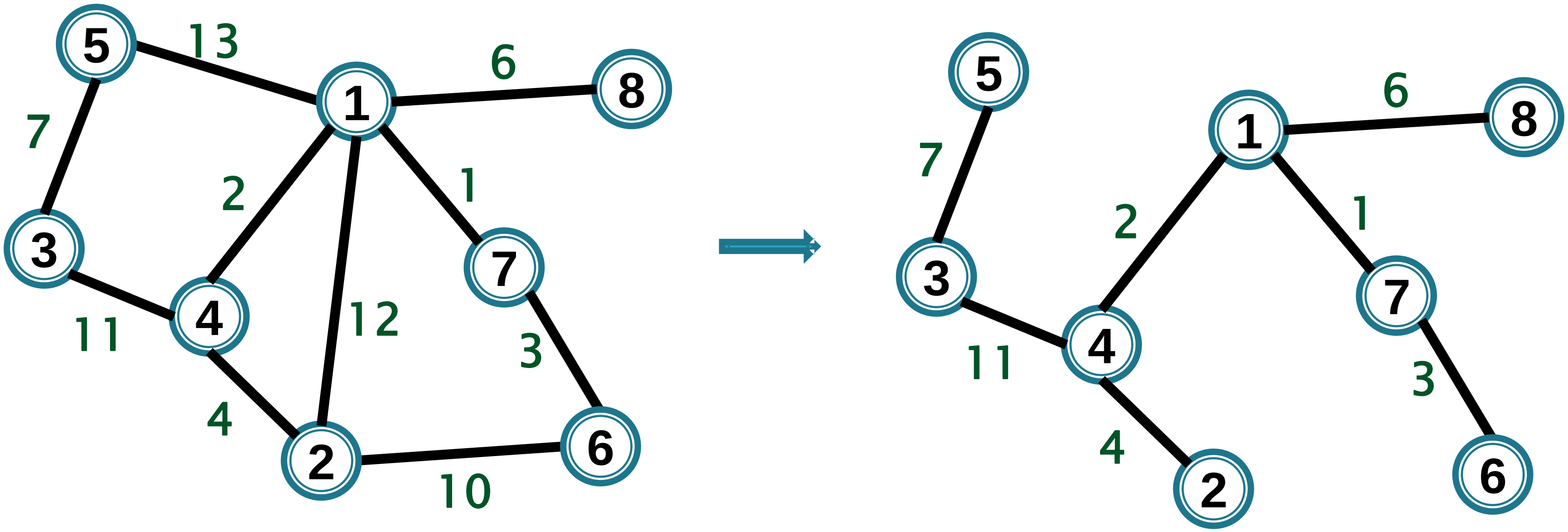


x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

(2,6): $w(2,6) - w(max[2,6]) = w(2,6) - w(2,4) = 10 - 4 = 6$

(1,2): $w(1,2) - w(max[1,2]) = w(1,2) - w(2,4) = 12 - 4 = 8$

(1,5): $w(1,5) - w(max[1,5]) = w(1,5) - w(4,3) = 13 - 11 = 2$



x	1	2	3	4	5	6	7	8
max[1,x]	0	(4,2)	(4,3)	(1,4)	(4,3)	(7,6)	(1,7)	(1,8)
max[2,x]	(2,4)	0	(3,4)	(2,4)	(3,4)	(2,4)	(2,4)	(1,8)

(2,6): w(2,6) – w(max[2,6]) = w(2,6) – w(2,4) = 10 – 4 = 6

(1,2): w(1,2) – w(max[1,2]) = w(1,2) – w(2,4) = 12 – 4 = 8

(1,5): w(1,5) – w(max[1,5]) = w(1,5) – w(4,3) = 13 – 11 = 2

=> **Second best = T_{min} – max[1,5] + (1,5) = T_{min} – (3,4) + (1,5)**

Second best

Algoritm second best

1. Determinăm T apcm în G

2. Pentru orice $x, y \in T$ determină:

    $max[x,y]$ = **muchia** maximă din lanțul de la x la y din T

3. Determină o muchie $xy \notin T$ cu

    $w(x,y) - w(max[x,y])$ minim

4. $T_s = T + xy - max[x,y]$

Complexitate?

Second best

Algoritm second best

1. Determinăm T apcm în G

2. Pentru orice x, y $\in$ T determină:

   max[x,y] = **muchia** maximă din lanțul de la x la y din T

3. Determină o muchie xy $\notin$ T cu

   w(x,y) – w(max[x,y]) minim

4. $T_s = T + xy - max[x,y]$

Complexitate O(n²) – cu varianta descrisă la pasul 3

Alte idei pentru pasul 3?