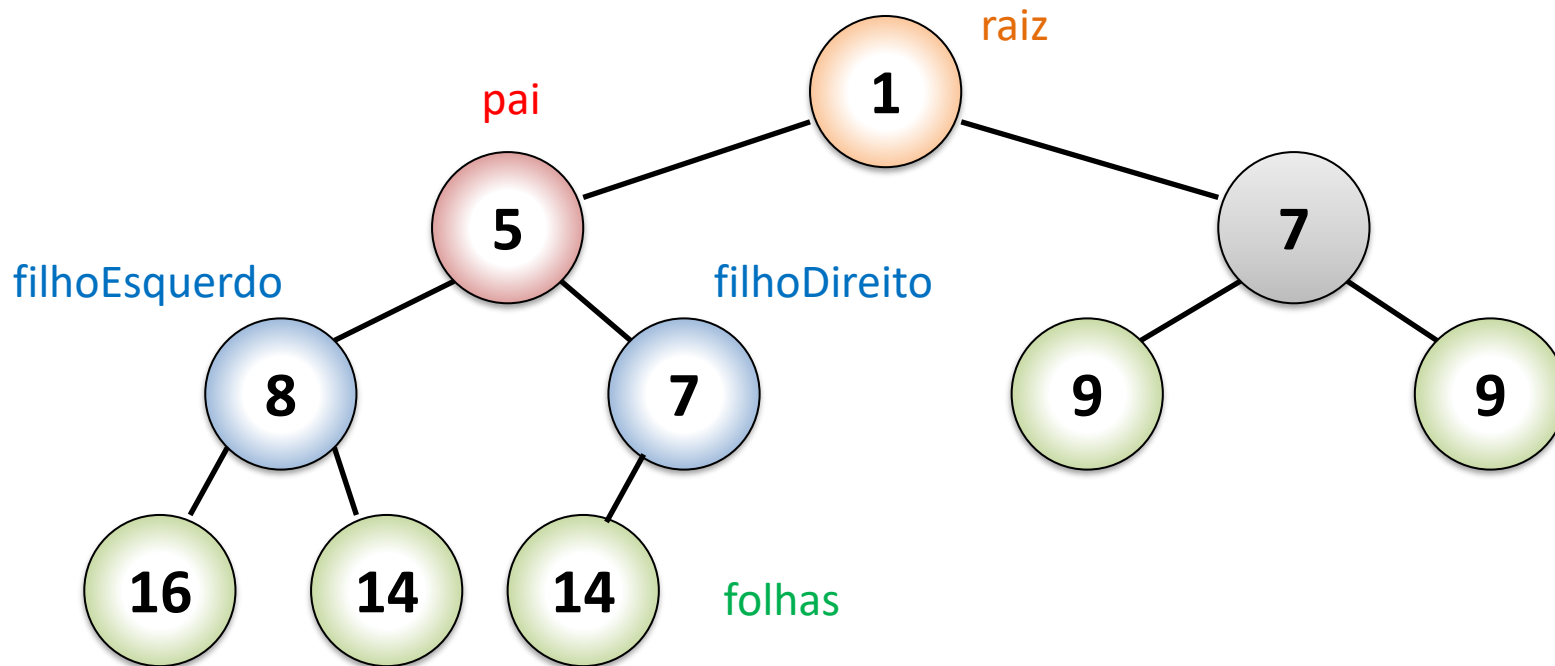


# Introdução à Teoria dos Grafos

Prof. Alexandre Noma

# Aula passada: (**min**) Heap

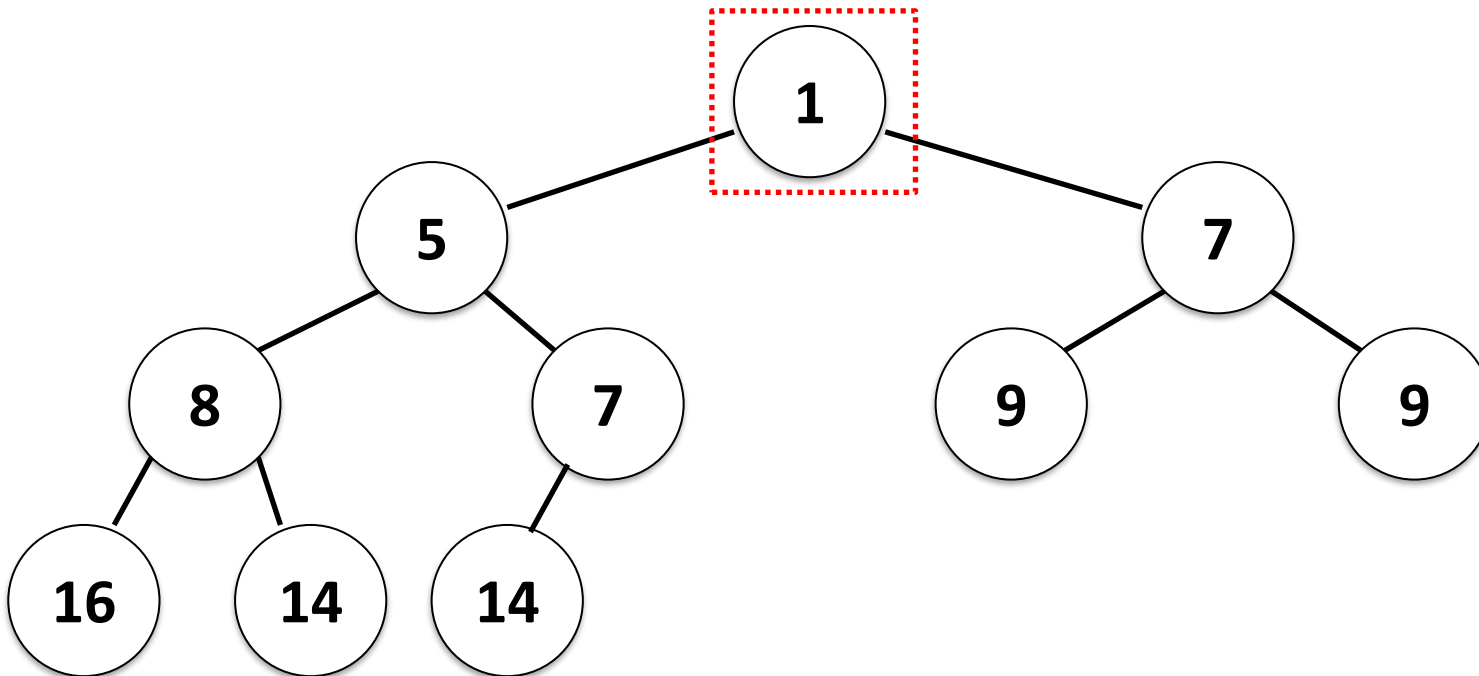
- Propriedade
  - **Chave** do **pai** é **menor ou igual** a dos seus **filhos**



# (min) Heap

vetor

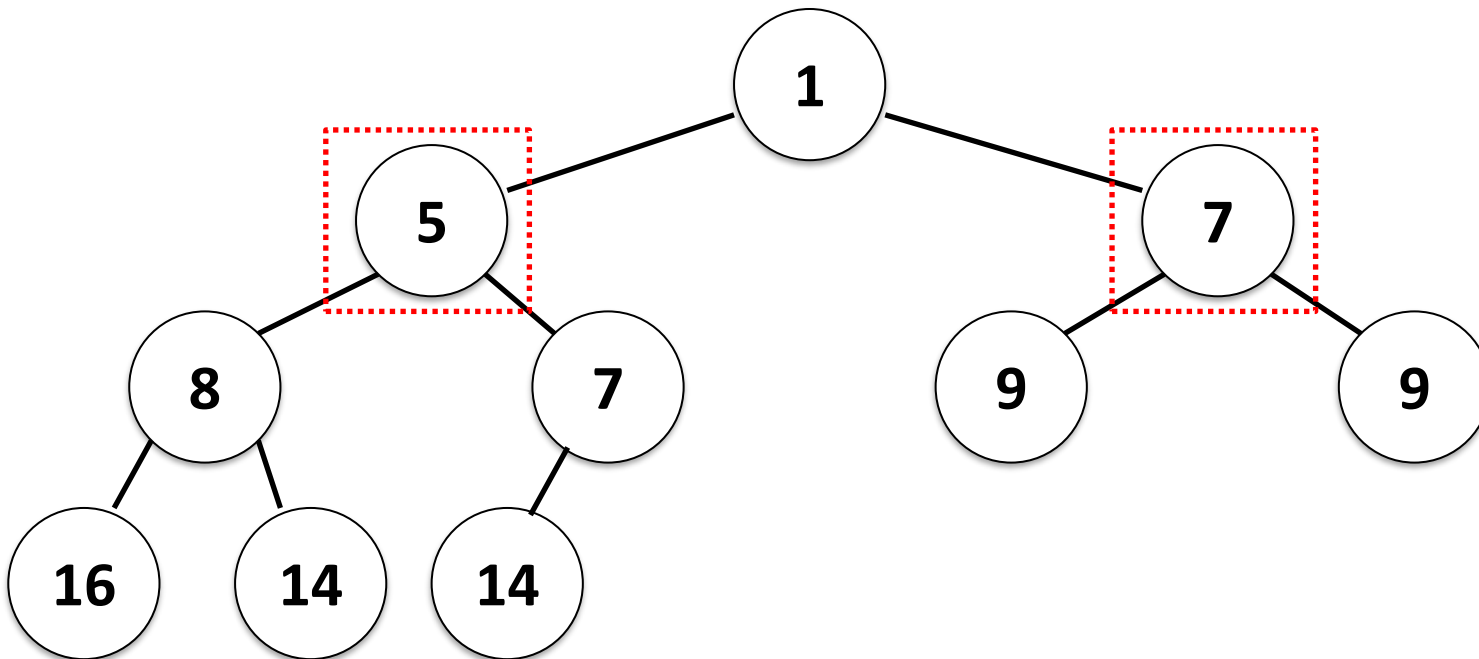
0	1	2	3	4	5	6	7	8	9
1	5	7	8	7	9	9	16	14	14



# (min) Heap

vetor

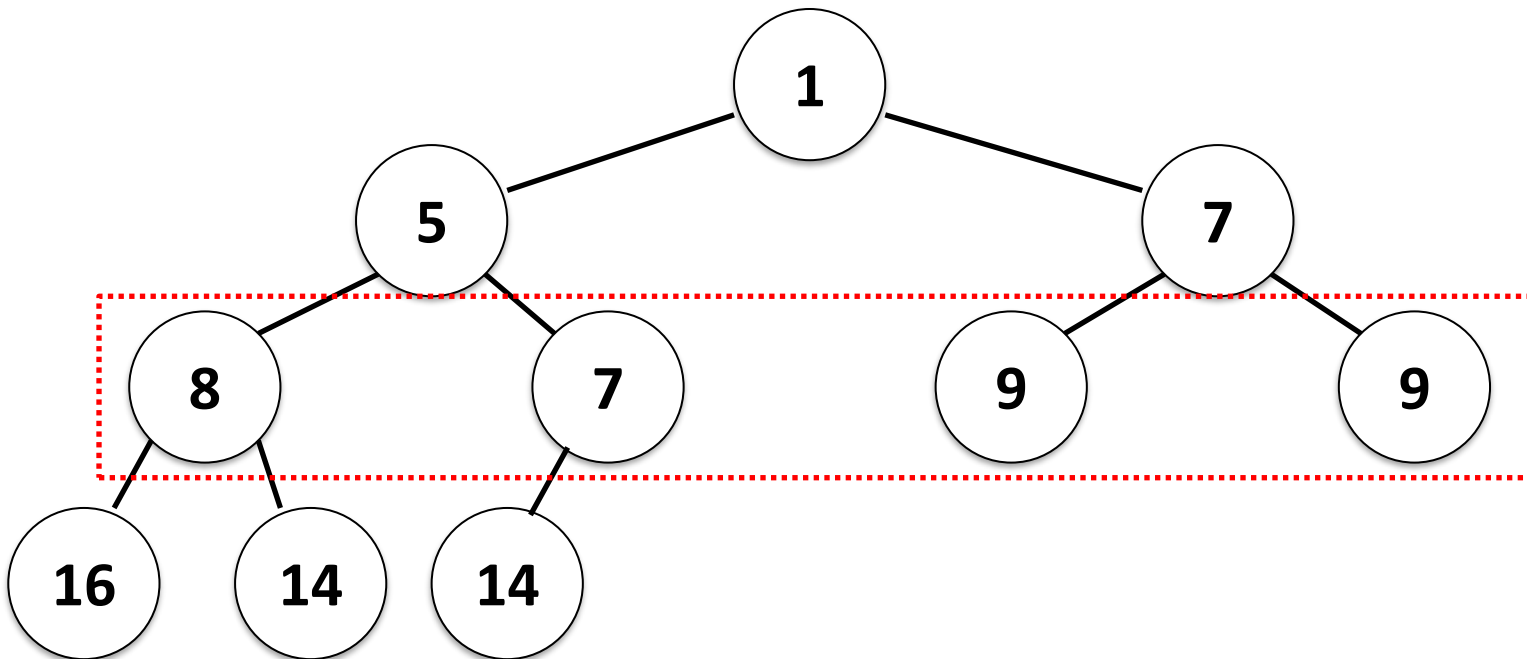
0	1	2	3	4	5	6	7	8	9
1	5	7	8	7	9	9	16	14	14



# (min) Heap

vetor

0	1	2	3	4	5	6	7	8	9
1	5	7	8	7	9	9	16	14	14



# Fila de **prioridade** (Heap)

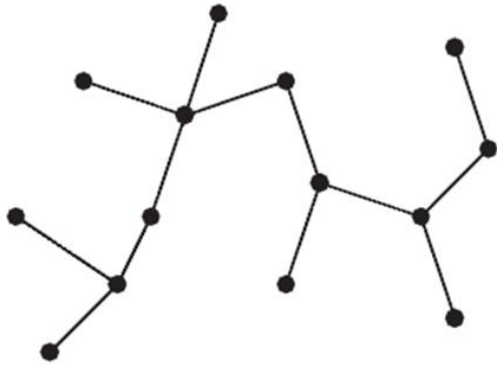
- **ConstroiHeap(Q)** consome  **$O(n)$**  unidades de tempo.
  - Constrói um Heap em um vetor
- **ExtraiMínimo(Q)** consome  **$O(\log n)$**  unidades de tempo.
  - remove e devolve o elemento de Q com a **menor chave**
- **Vazio(Q)** consome  **$O(1)$**  unidades de tempo.
  - devolve verdadeiro se fila vazia, falso caso contrário
- **DiminuiChave(Q, x, k)** consome  **$O(\log n)$**  unidades de tempo.
  - diminui o valor da chave de x para o novo valor k.

# Hoje

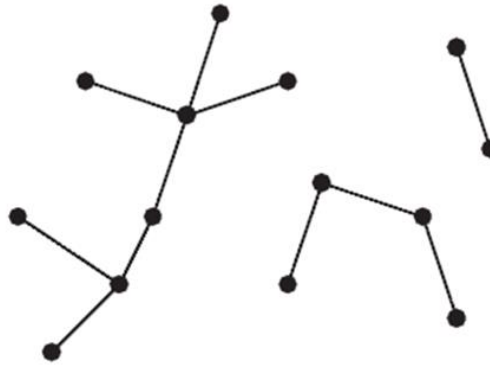
- Árvore geradora mínima
- Definições:
  - Árvore
  - Floresta
  - Árvore geradora
  - Árvore geradora de peso mínimo

# Árvore

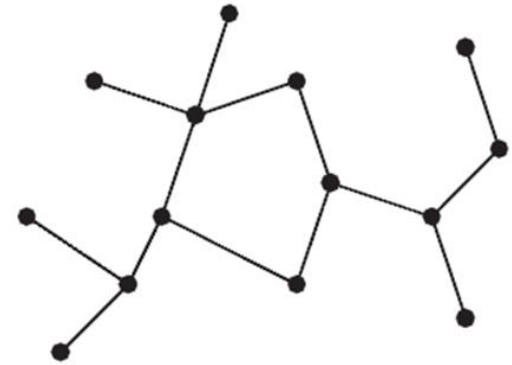
- Uma **árvore** é um grafo **conexo** e **acíclico** (sem ciclos).



(a)



(b)



(c)

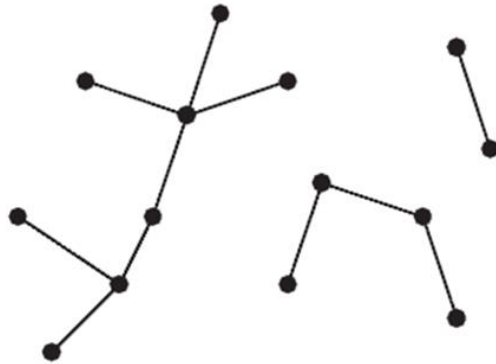


# Árvore

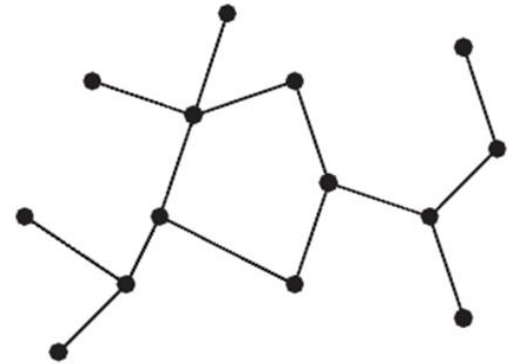
- Uma **árvore** é um grafo **conexo** e **acíclico** (sem ciclos).



(a)



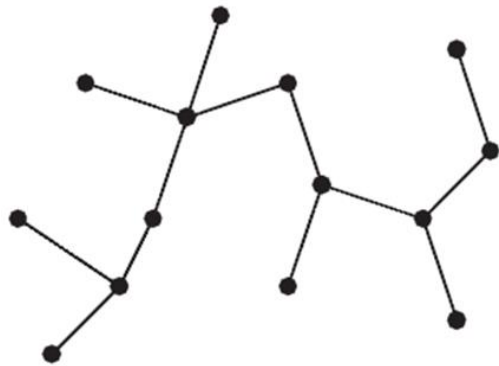
(b)



(c)

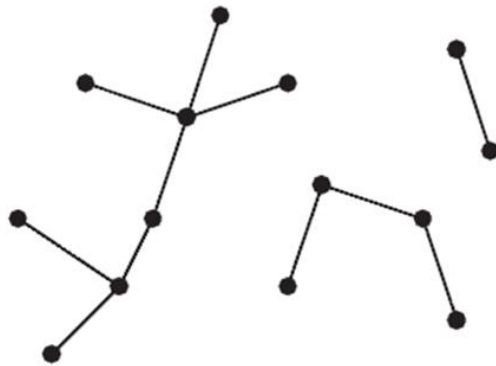


# Árvore vs Floresta



(a)

árvore



(b)

floresta



(c)



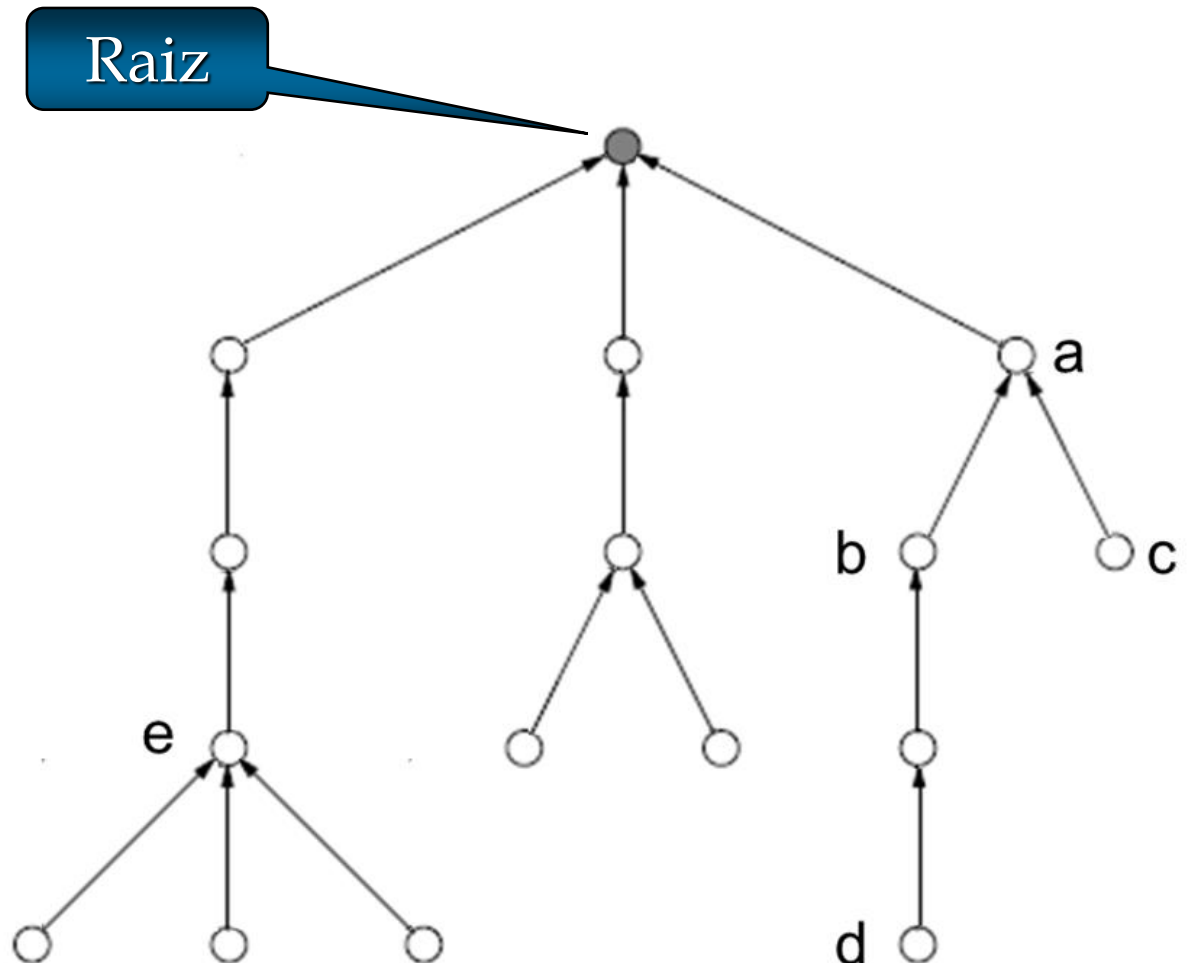
# Árvore enraizada

- Hierarquia

- pai
- filho
- irmão
- ancestral
- descendente

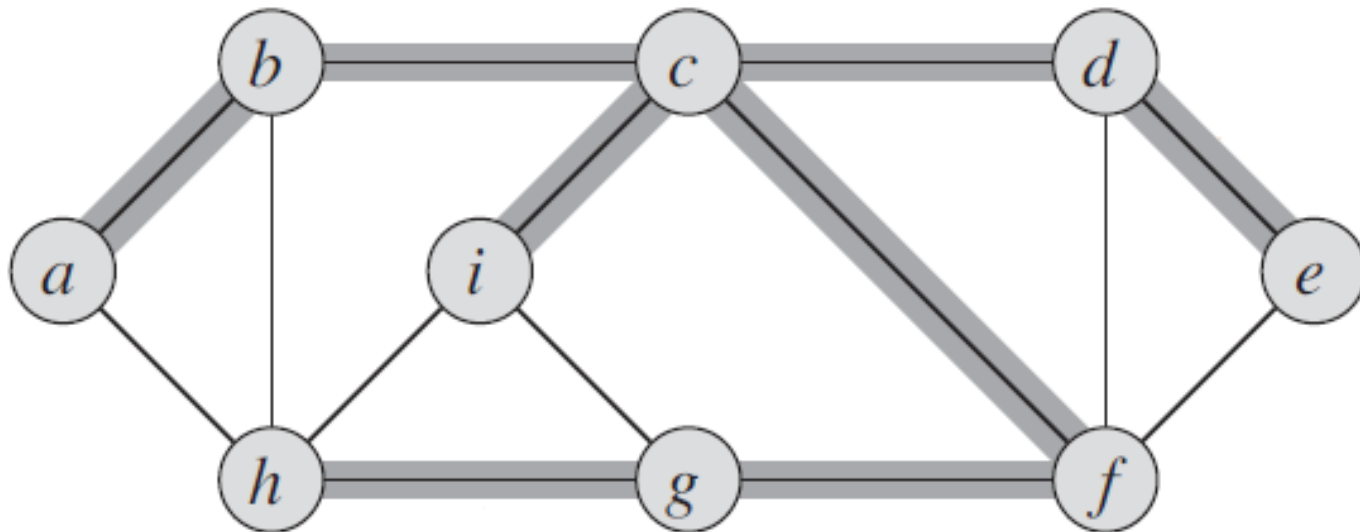
Raiz

Folha



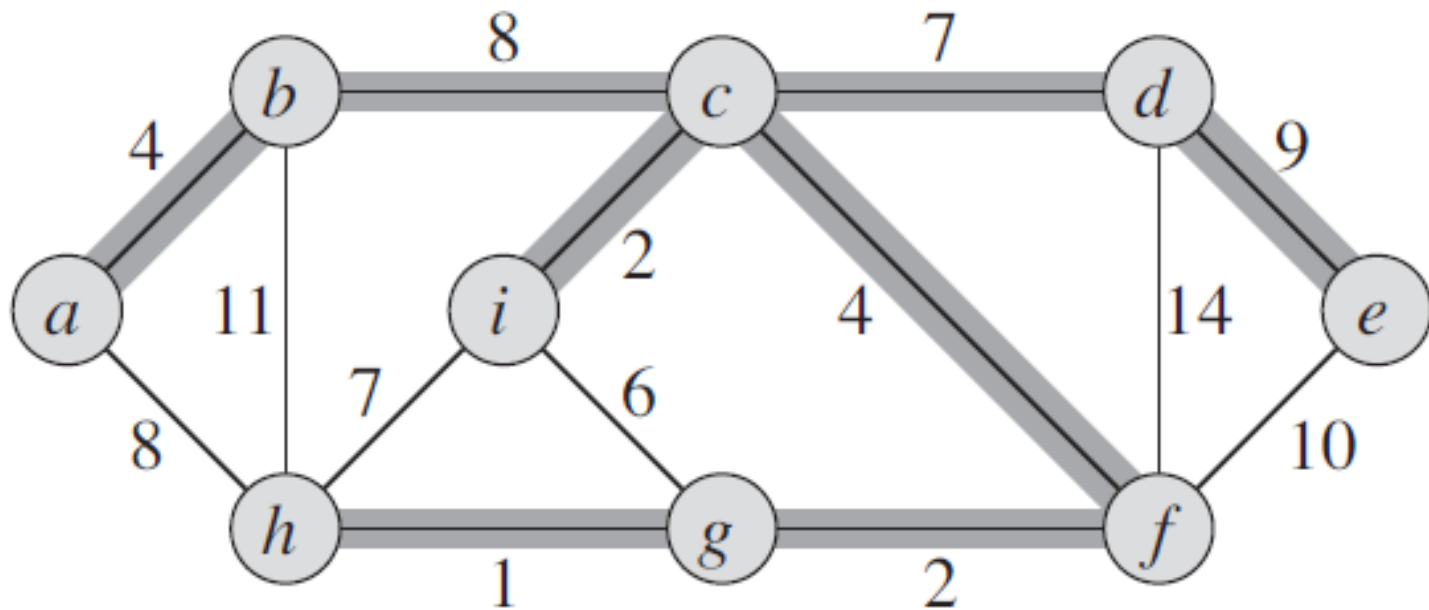
# Árvore geradora

- É uma árvore que conecta todos os vértices.



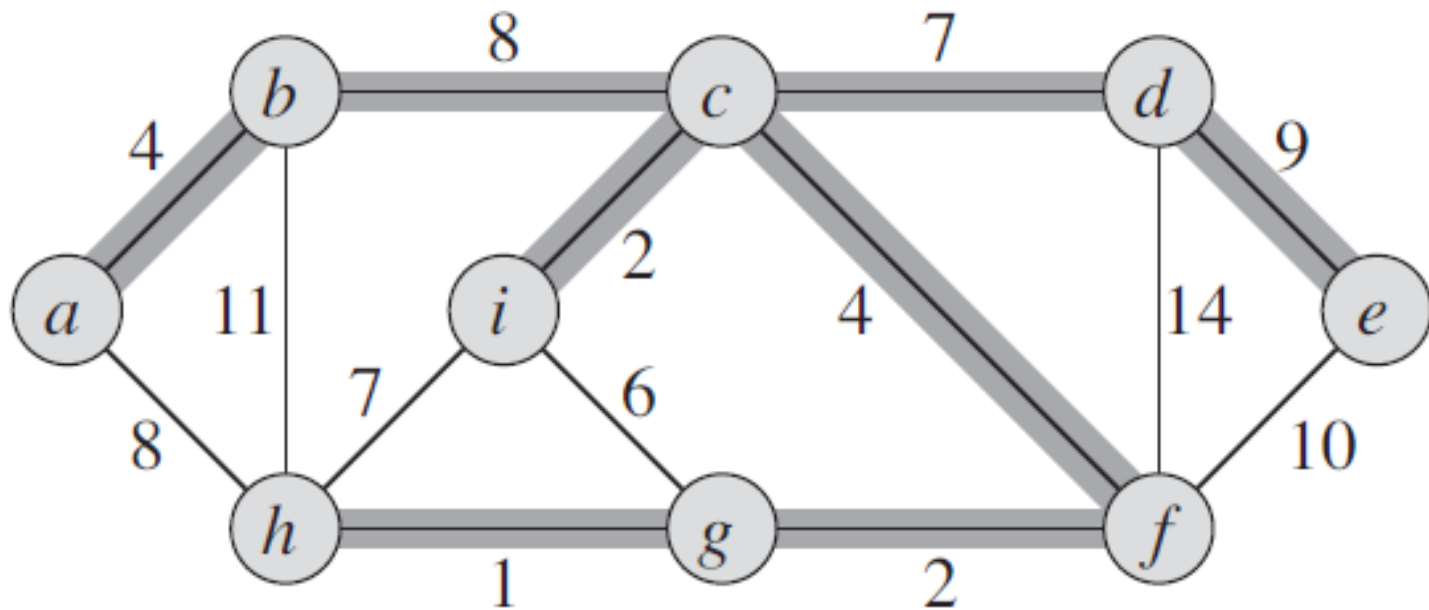
# Árvore **geradora mínima** ?

- É uma árvore que conecta todos os vértices...  
**???**



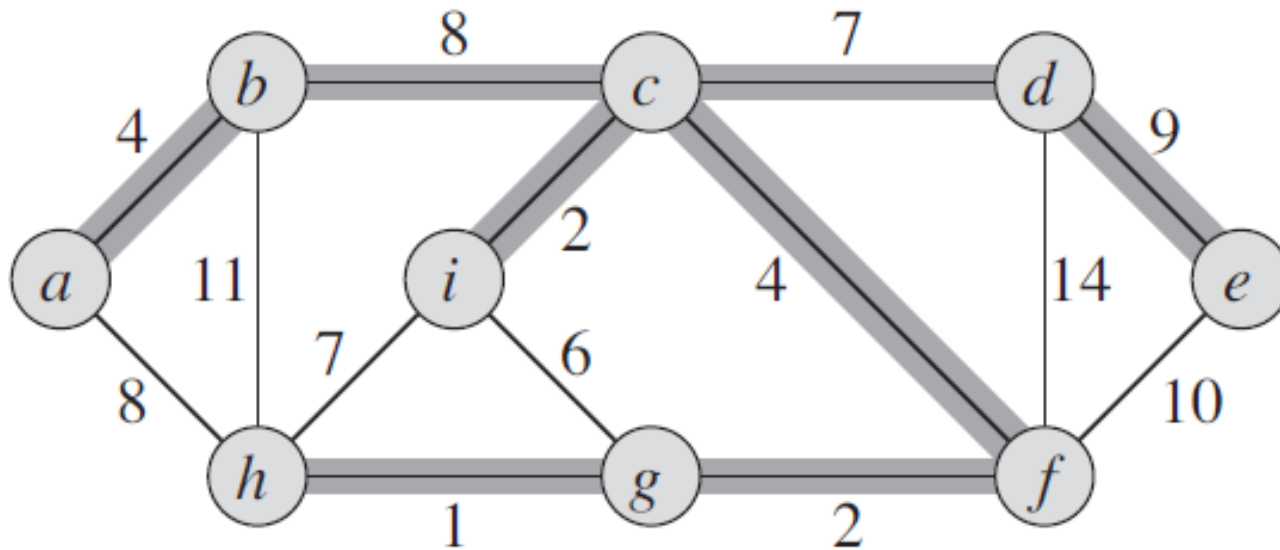
# Árvore geradora mínima ?

- É uma árvore que conecta todos os vértices de peso total mínimo.



# Problema: MST

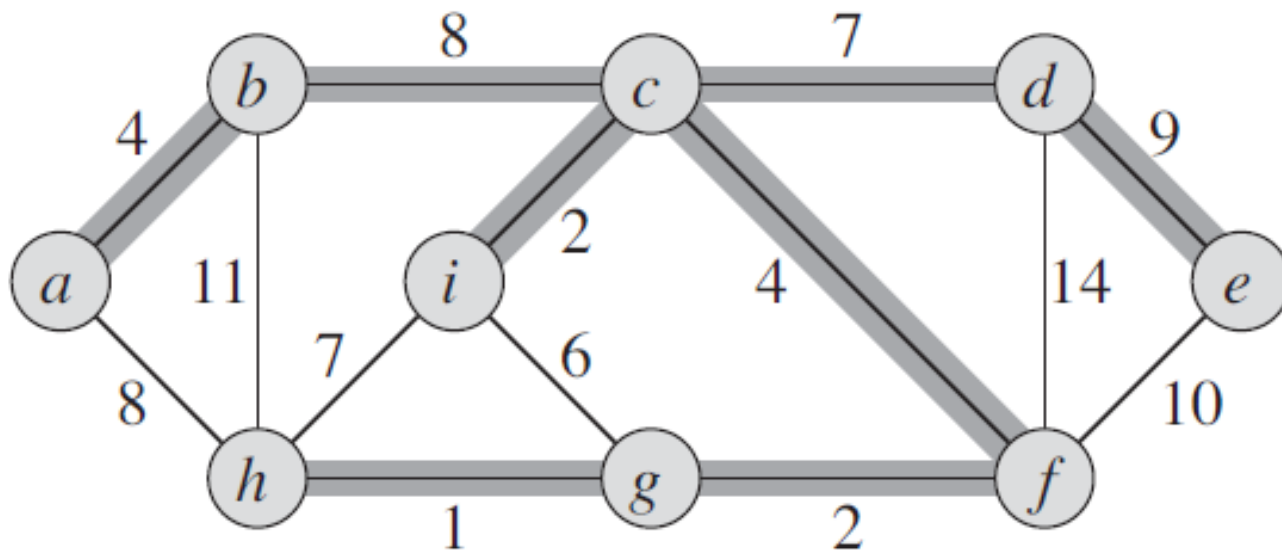
- Minimum Spanning Tree



Total:  $4+8+7+9+2+4+1+2 = 37$

# Problema: **MST**

- **Entrada:** Dado um grafo conexo **G**.
- **Saída:** Obter uma **árvore geradora mínima**.



Total:  $4+8+7+9+2+4+1+2 = 37$



# Algoritmo de Prim

- Prim( $G, w, r$ )
  - Entrada: um grafo conexo  **$G$** ,  
ponderado com pesos  **$w$** ,  
e um vértice inicial  **$r$**  (raiz).
  - Saída: **árvore geradora mínima**.
- Atributos
  - v.**chave**
  - v.**pai**

# Fila de **prioridade**?



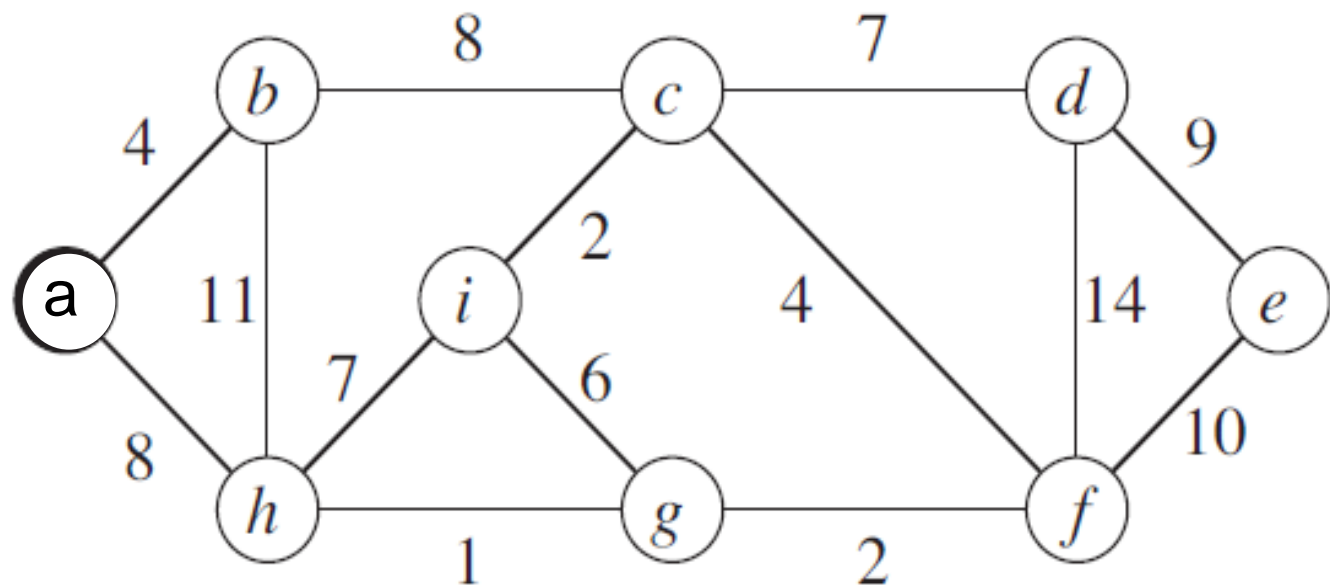
# Exercício Programa

- 09-filaDePrioridade.py  
(implementação simples e ineficiente com **vetor de índices**)
- Ex. Implementação eficiente com **HEAP**

# Fila de **prioridade** (Heap)

- **ConstroiHeap(Q)** consome  $O(n)$  unidades de tempo.
  - Constrói um Heap em um vetor
- **ExtraiMínimo(Q)** consome  $O(\log n)$  unidades de tempo.
  - remove e devolve o elemento de Q com a **menor chave**
- **Vazio(Q)** consome  $O(1)$  unidades de tempo.
  - devolve verdadeiro se fila vazia, falso caso contrário
- **DiminuiChave(Q, x, k)** consome  $O(\log n)$  unidades de tempo.
  - diminui o valor da chave de x para o novo valor k.

# Exemplo: MST



**MST-Prim**( $G, w, r$ )

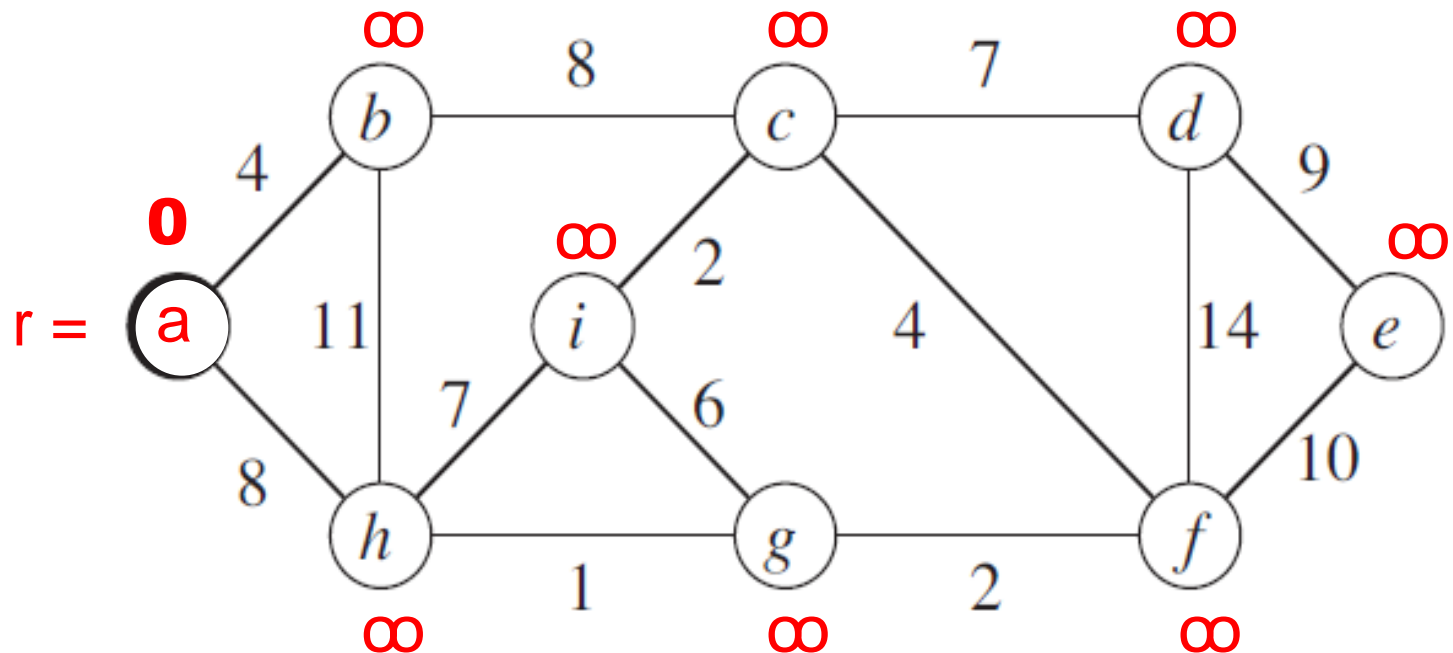
```
1 para cada vértice  $u$  em  $G.V$  faça
2      $u.chave = \text{INFINITO}$ 
3      $u.pai = \text{NIL}$ 
4  $r.chave = 0$ ;  $Q = G.V$ 


---


5 enquanto  $Q \neq \text{VAZIO}$  faça
6      $u = \text{ExtraiMinimo}(Q)$ 
7     para cada  $v$  em  $G.Adj[u]$  faça
8         se  $v$  está em  $Q$  e  $w(uv) < v.chave$ 
9             então  $v.chave = w(uv)$ ;  $v.pai = u$ 
```

## MST-Prim( $G, w, r$ )

- 1 para cada vértice  $u$  em  $G.V$  faça
  - 2      $u.\mathbf{chave} = \text{INFINITO}$
  - 3      $u.\mathbf{pai} = \text{NIL}$
  - 4  $r.\mathbf{chave} = 0$ ;     $Q = G.V$
- 



## MST-Prim( $G, w, r$ )

...

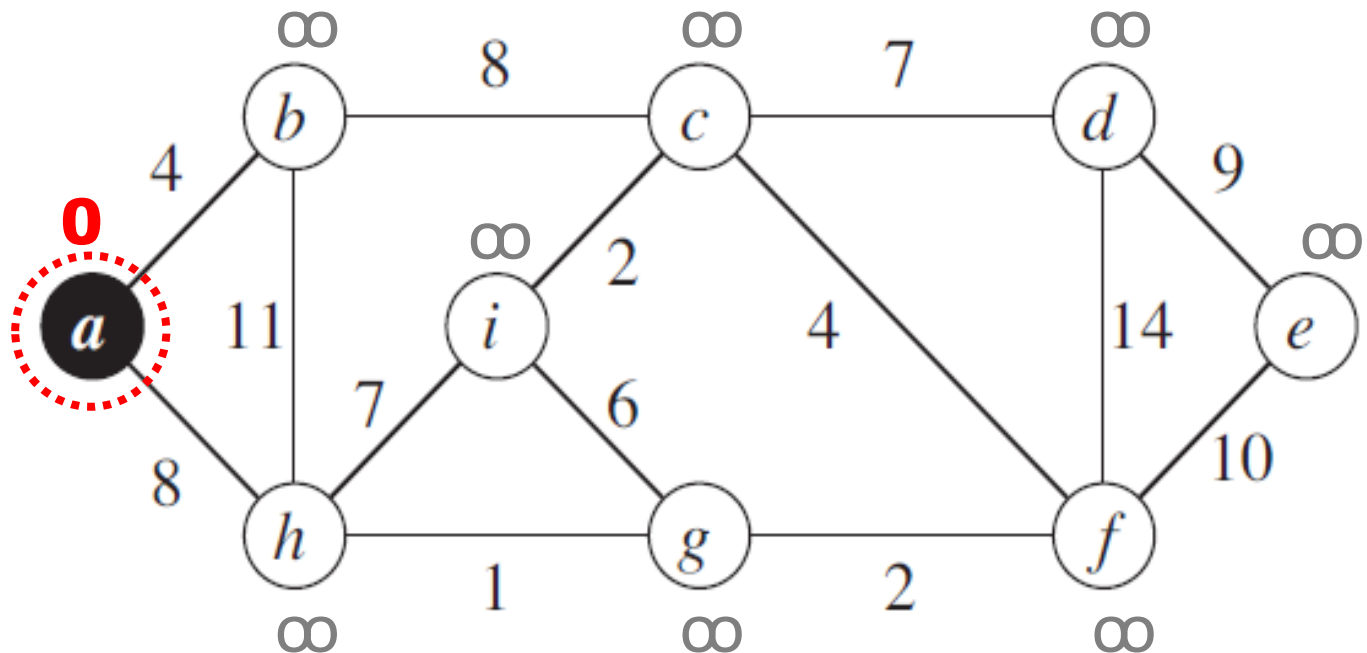
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$





## MST-Prim( $G, w, r$ )

...

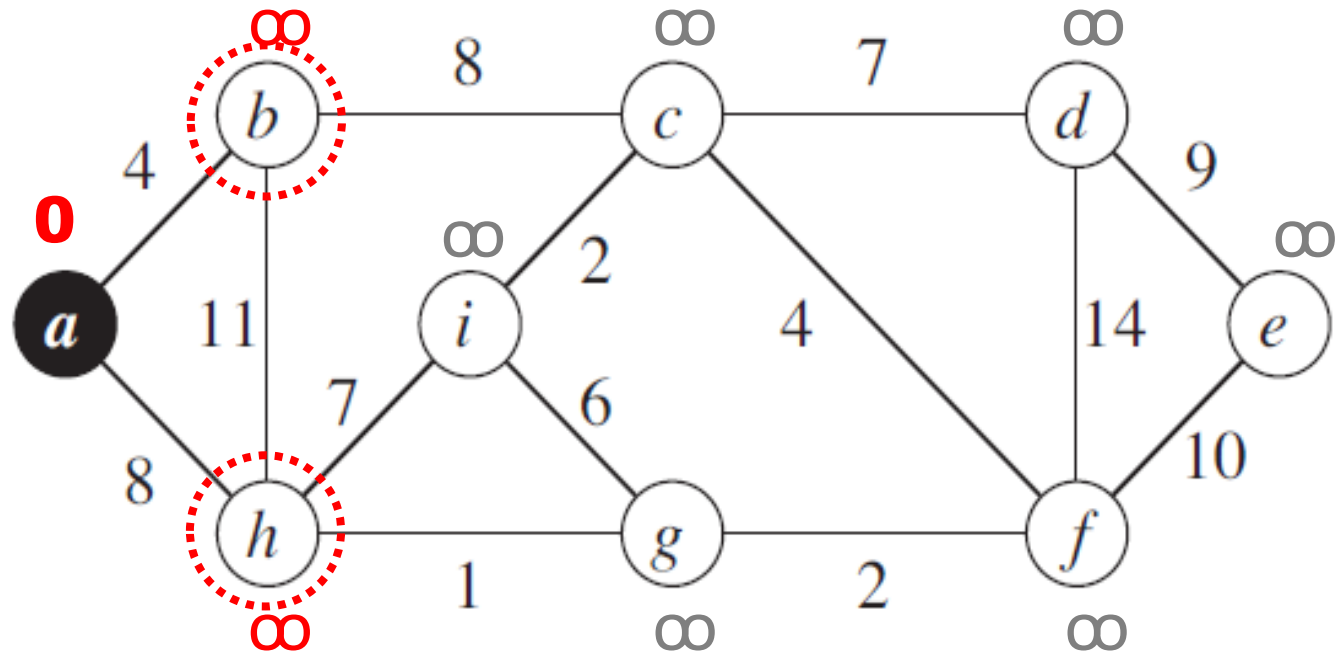
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

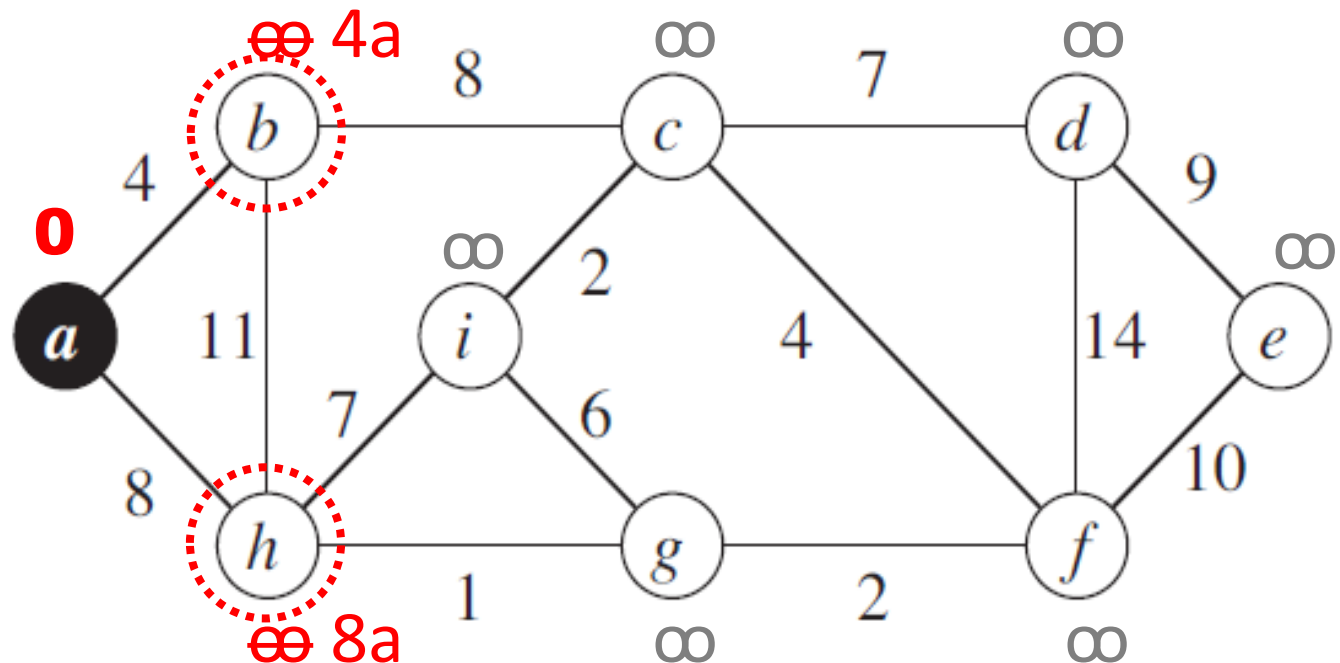
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

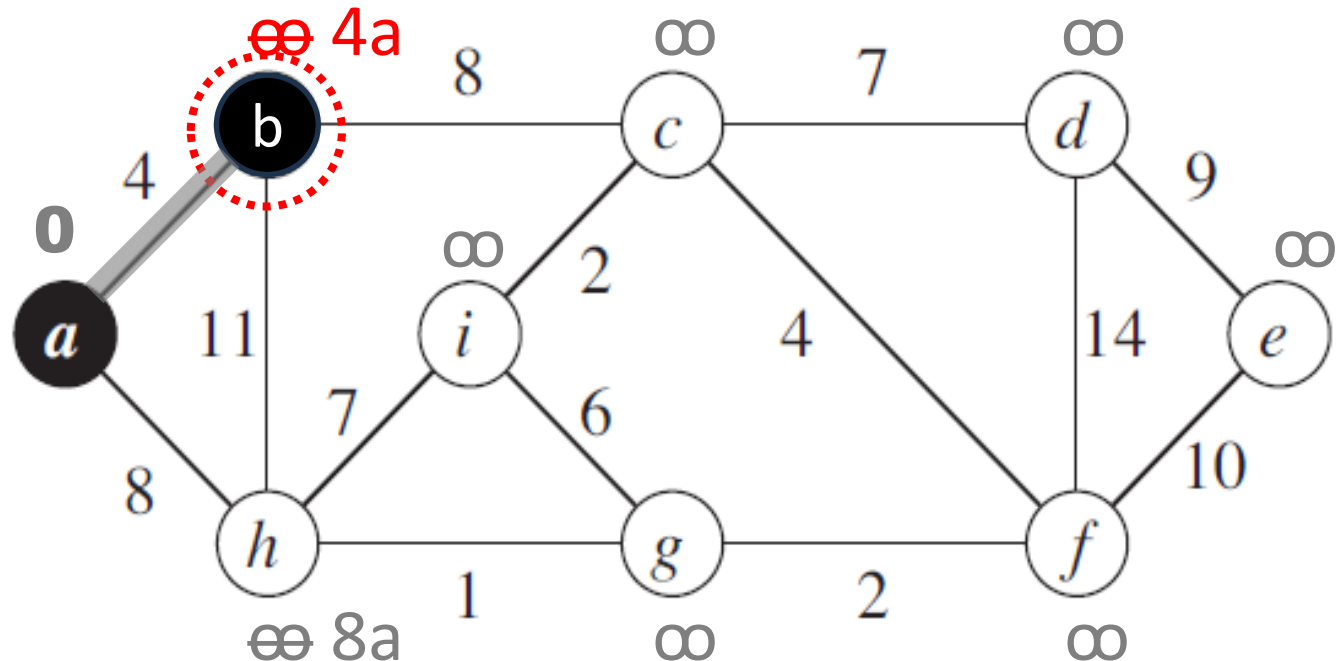
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

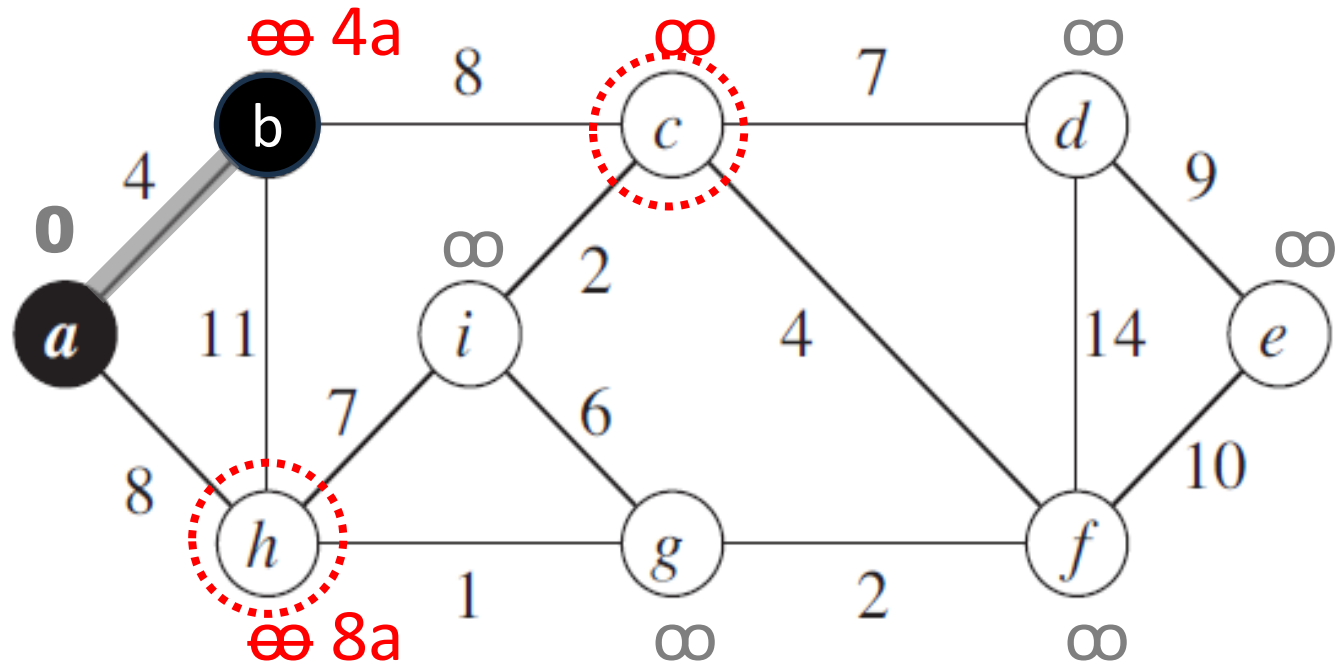
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

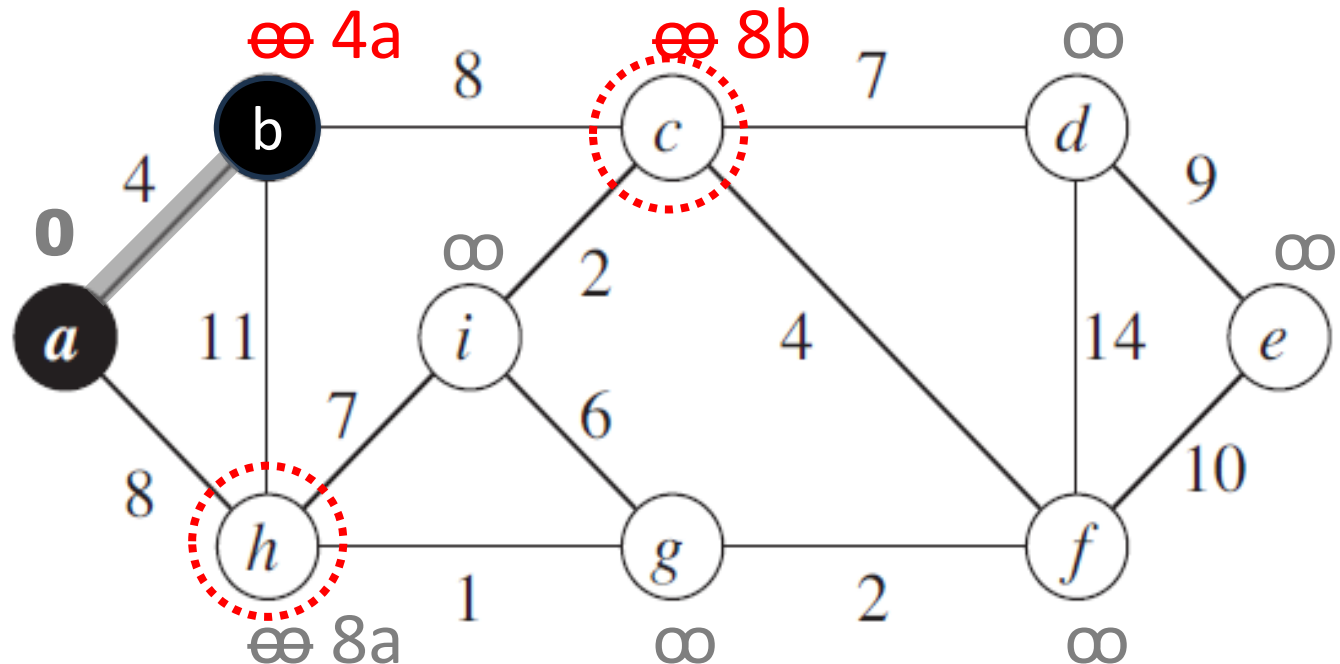
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

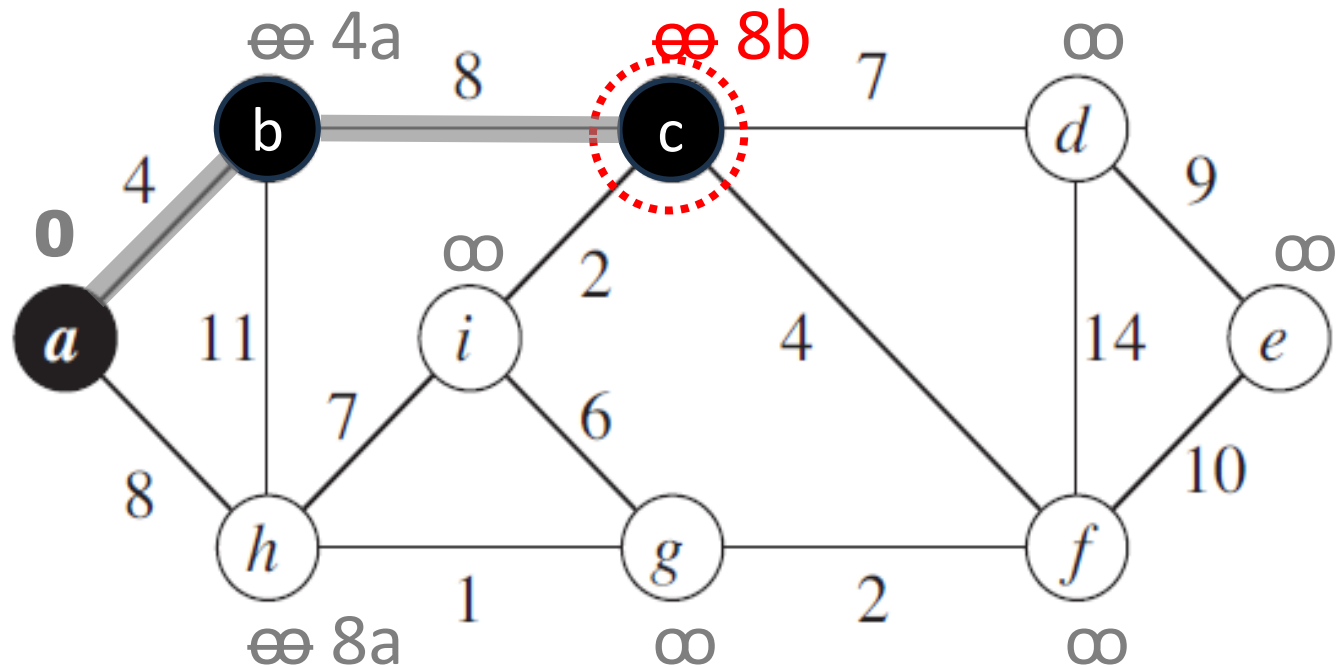
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

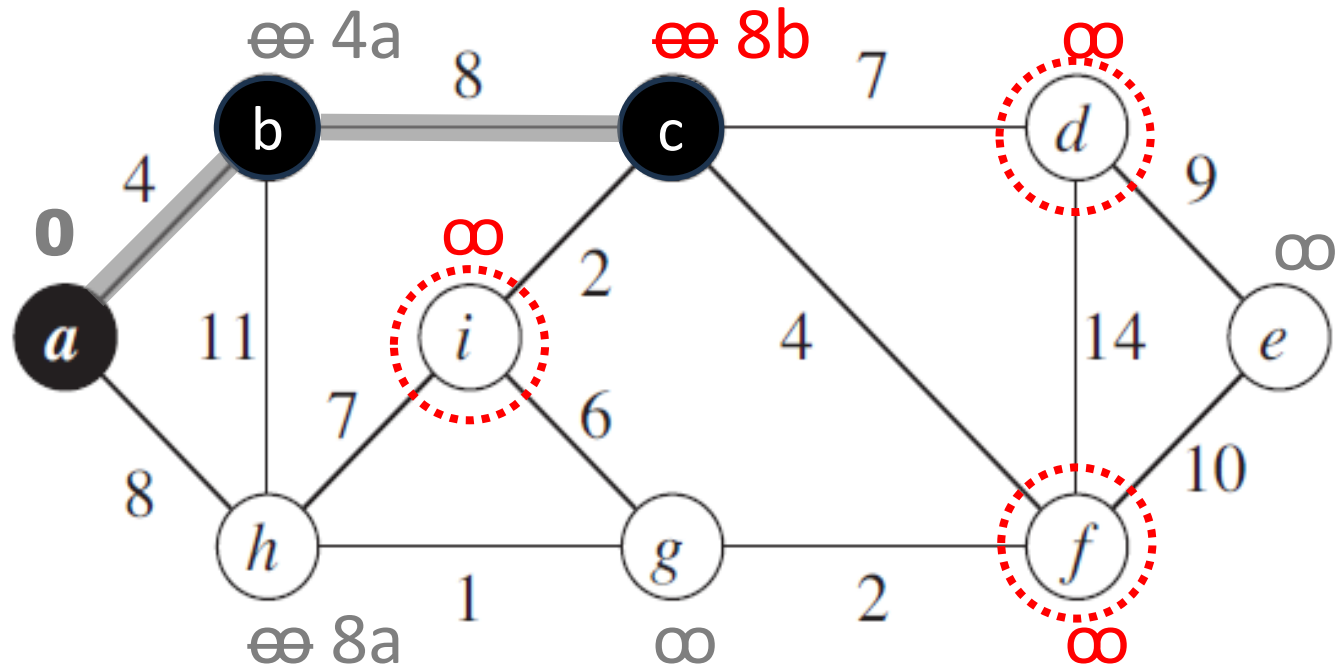
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

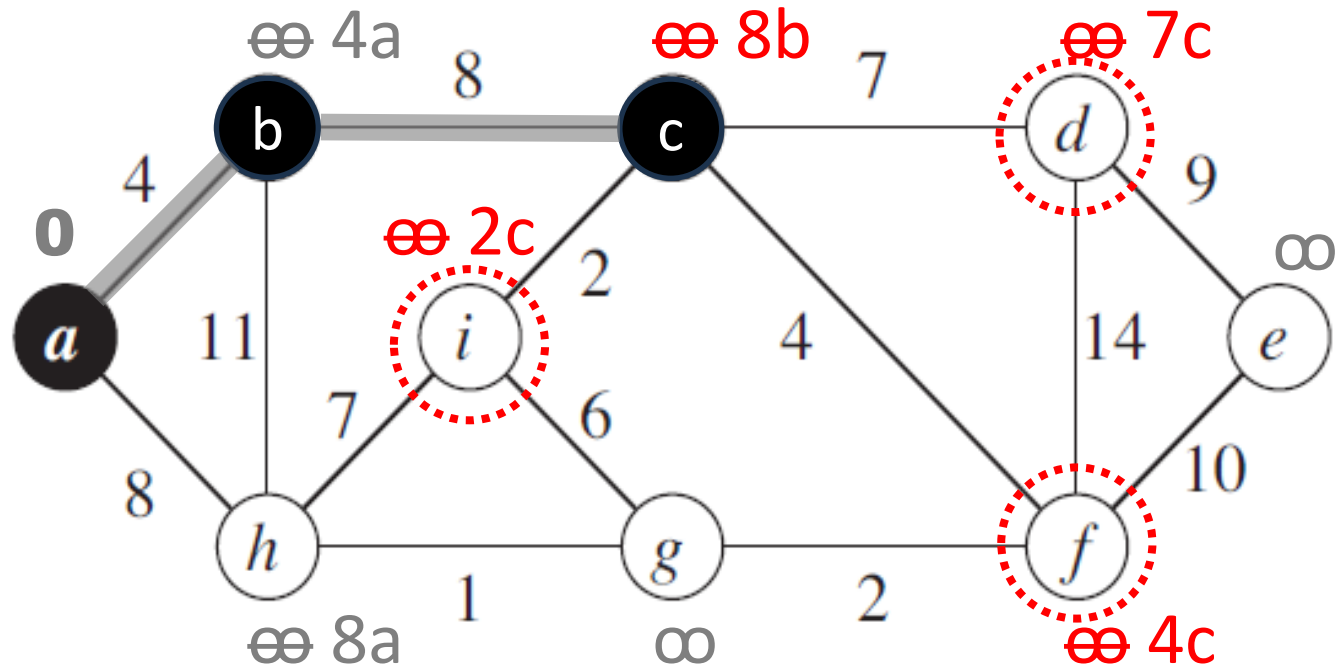
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$





## MST-Prim( $G, w, r$ )

...

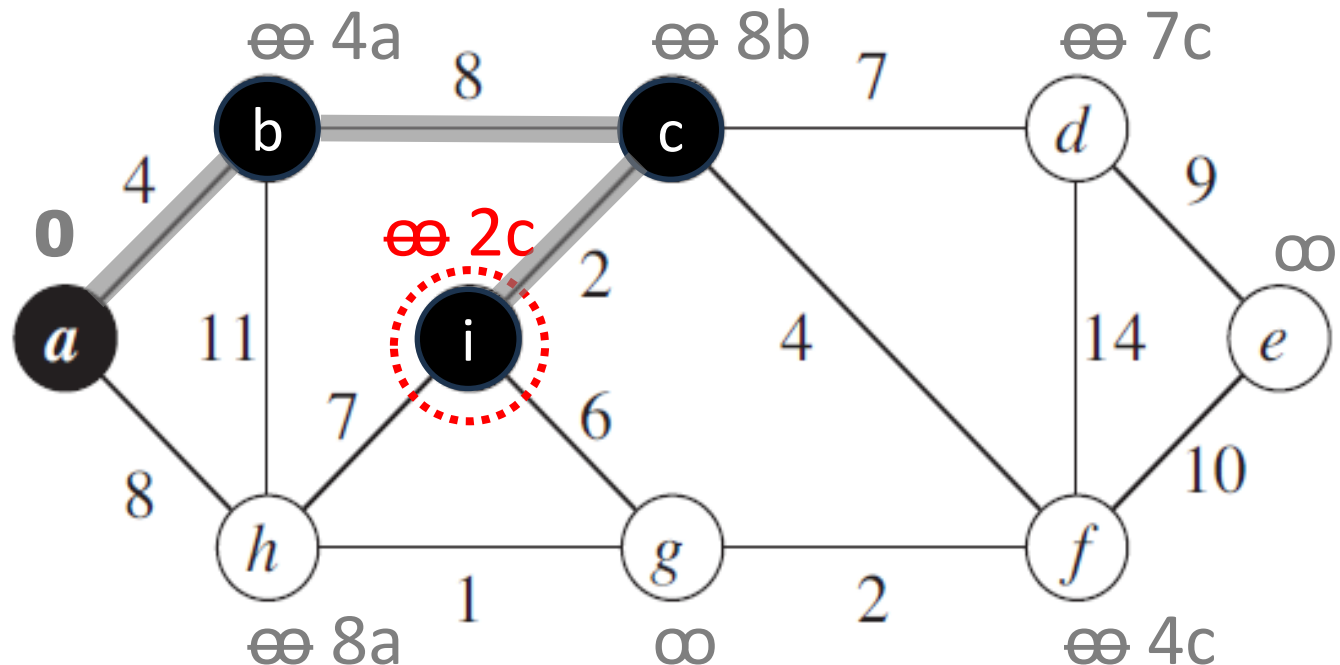
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

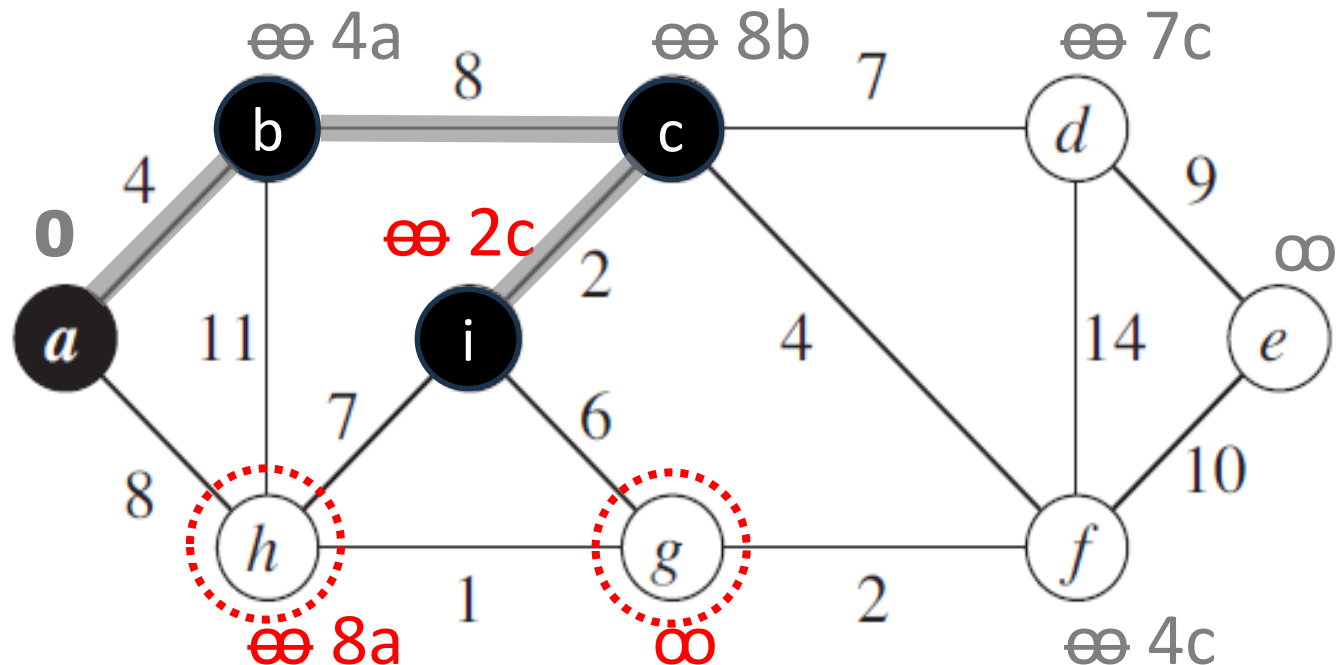
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

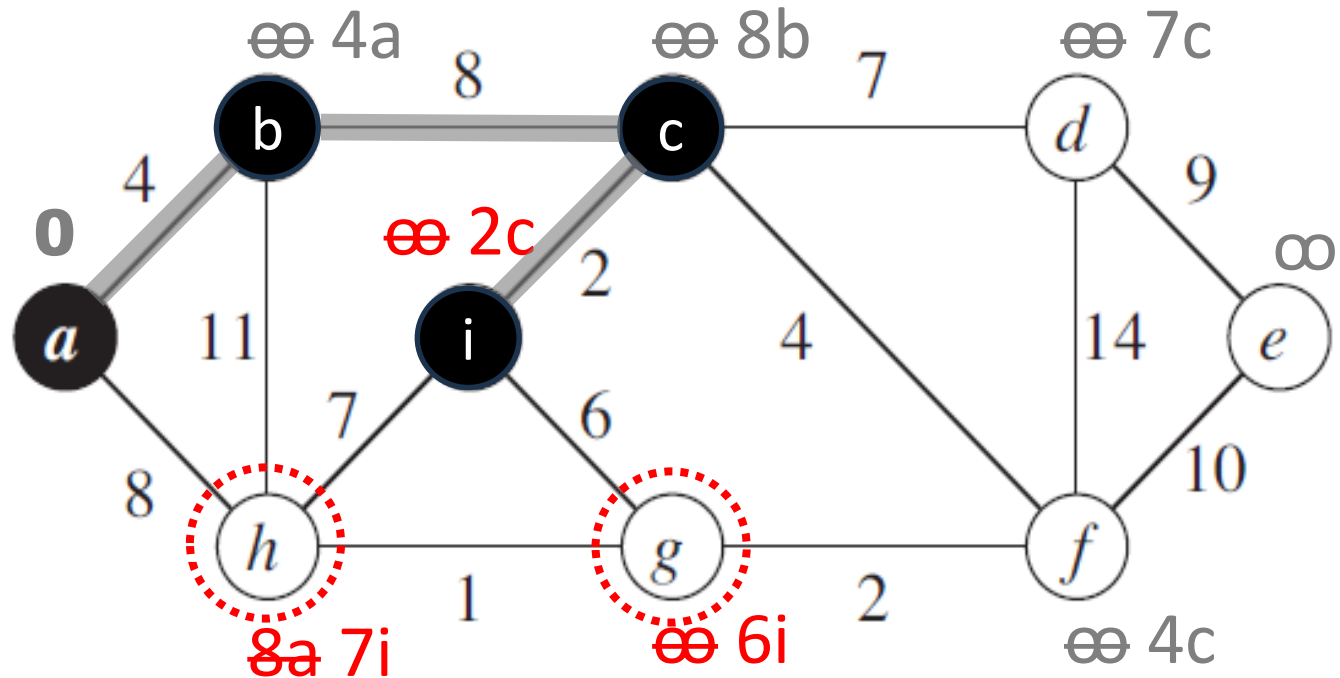
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

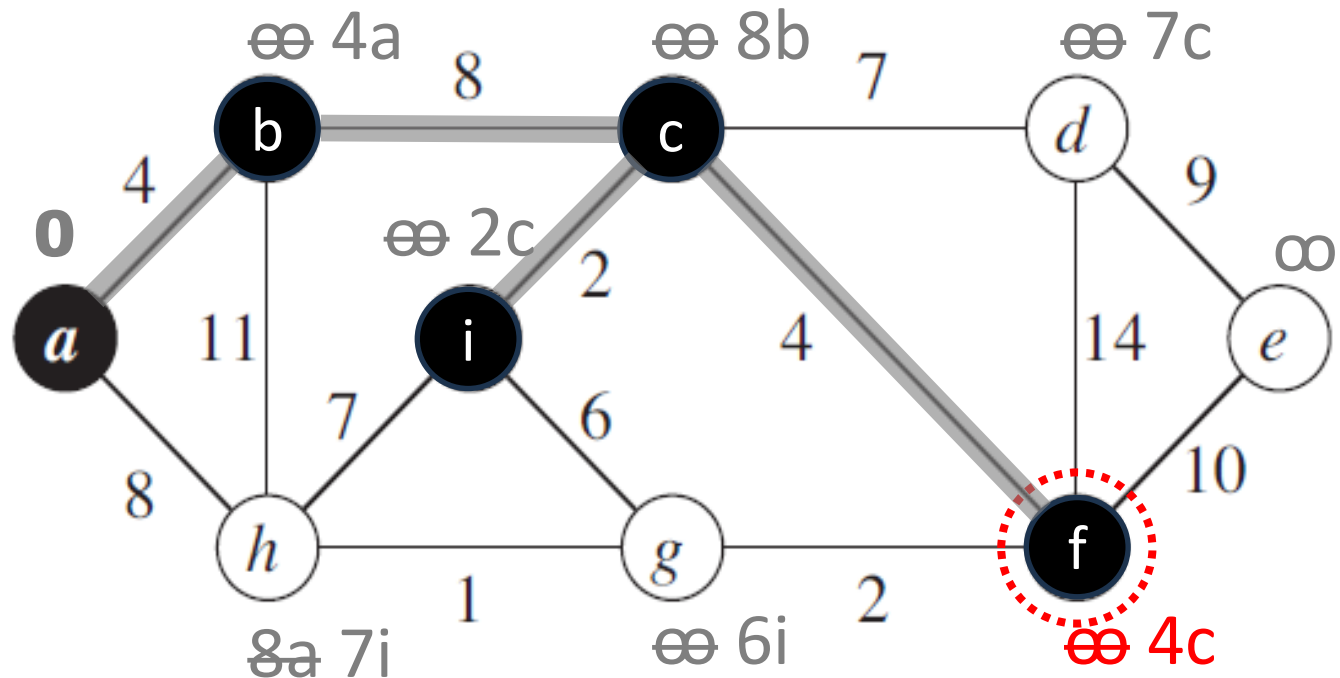
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

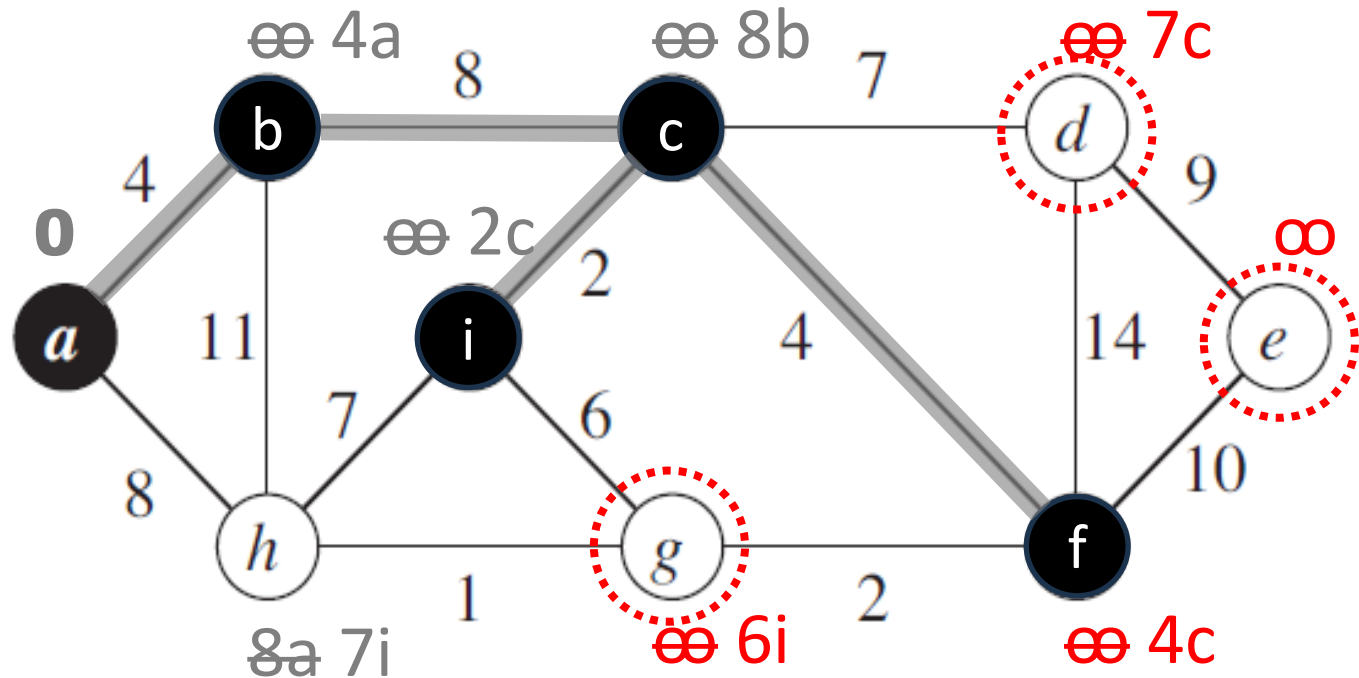
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

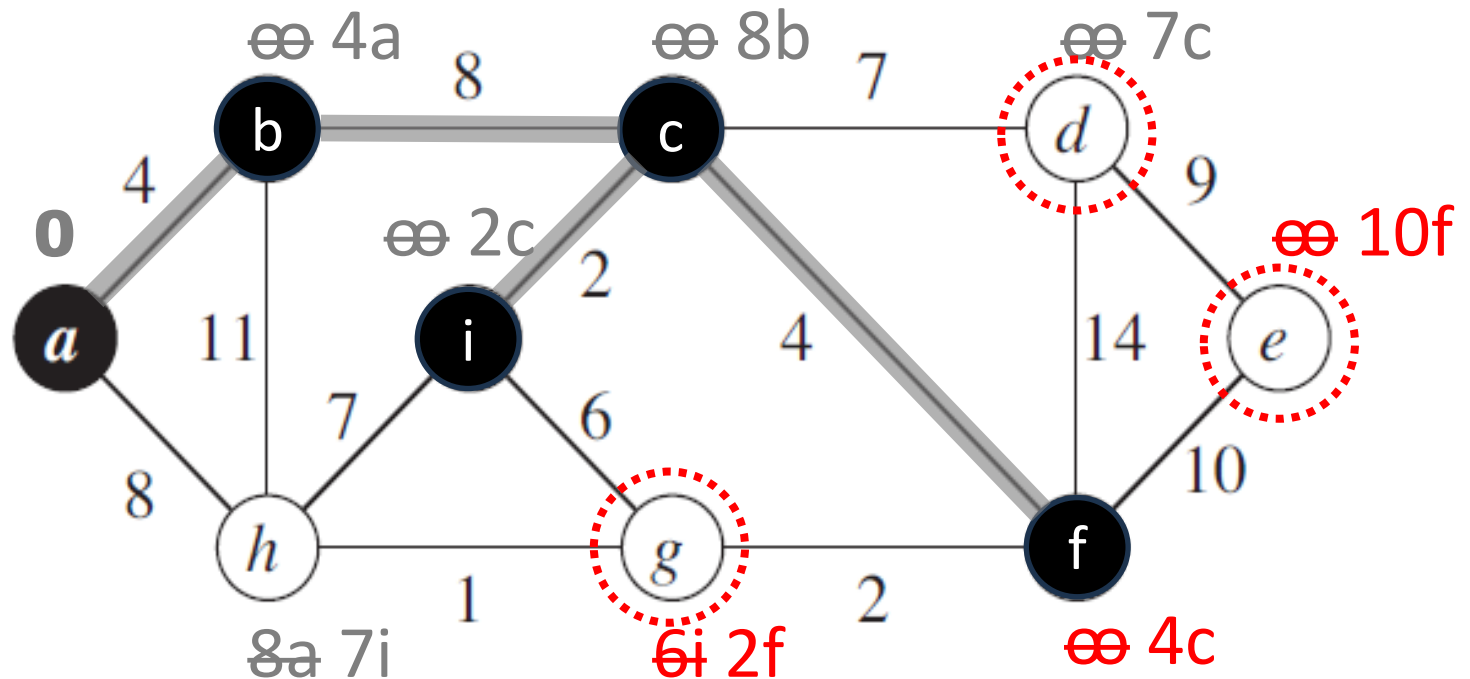
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

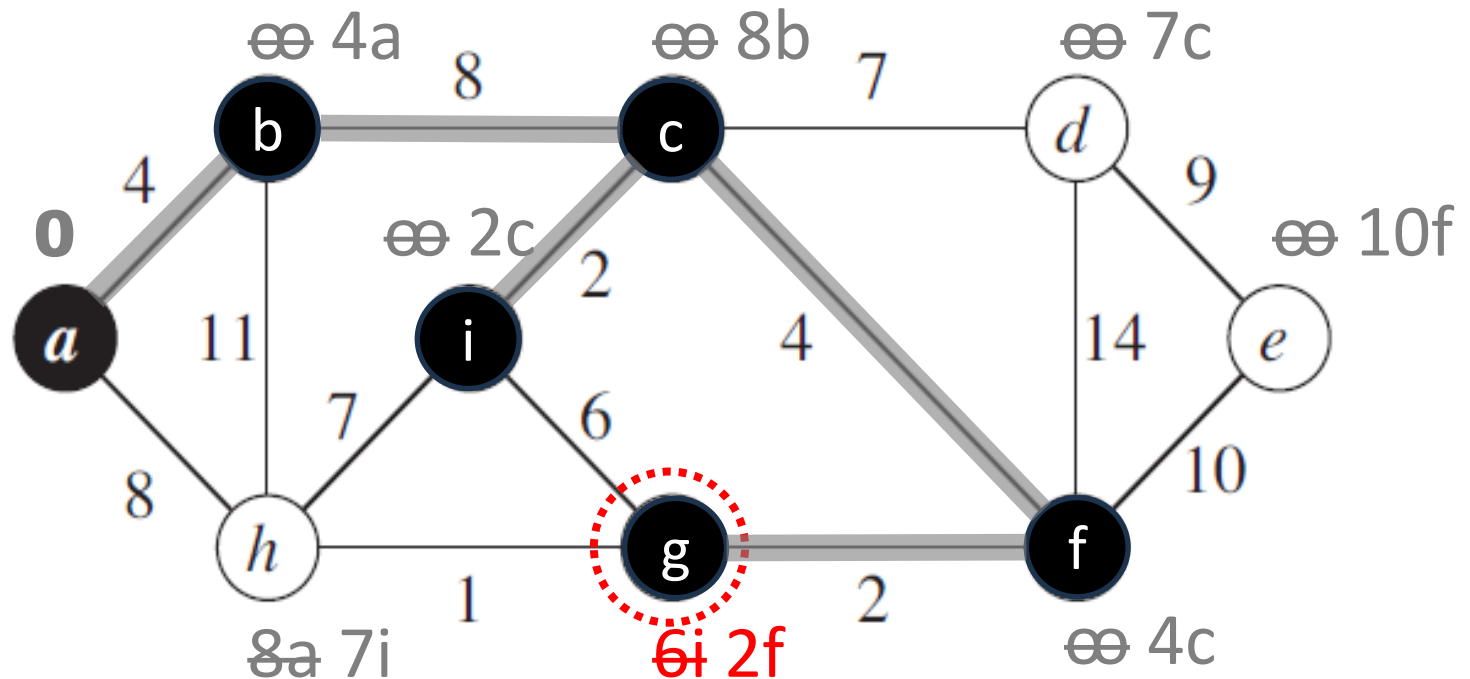
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

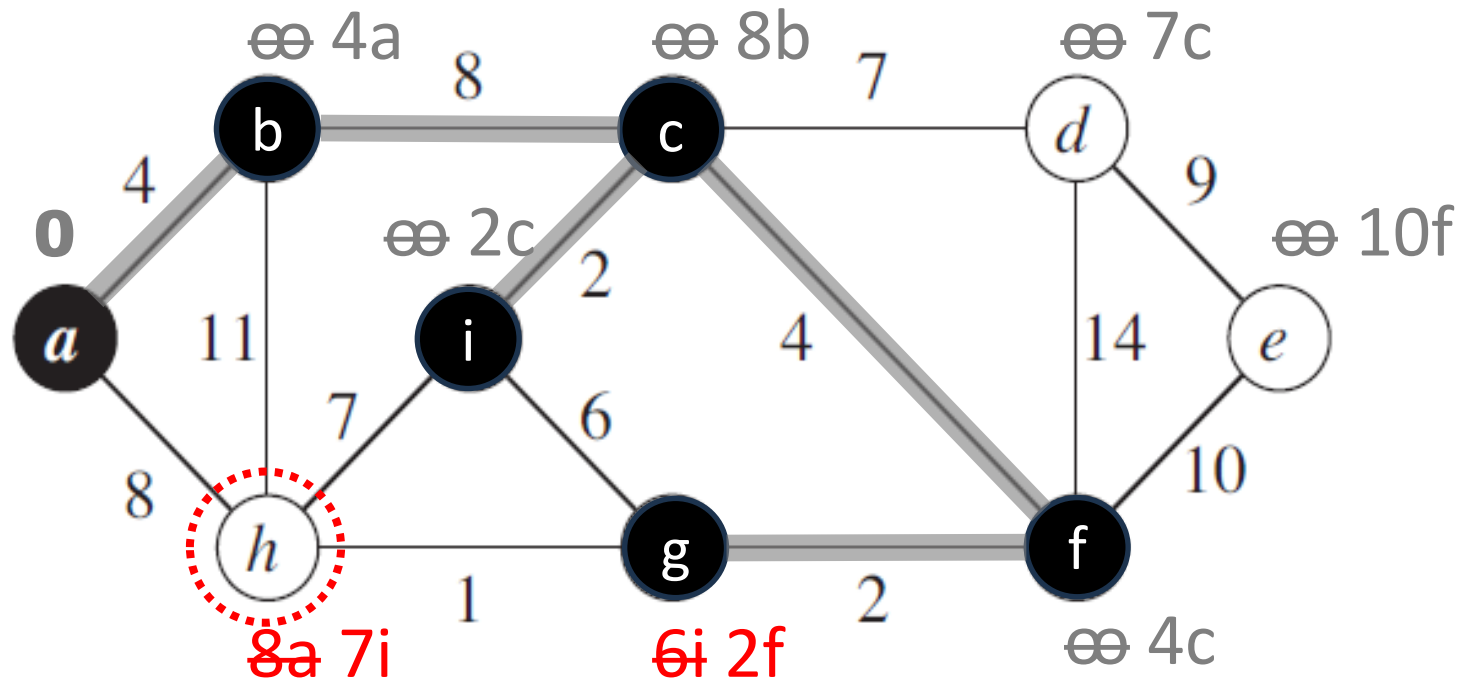
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  **$v$  está em  $Q$**  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$





## MST-Prim( $G, w, r$ )

...

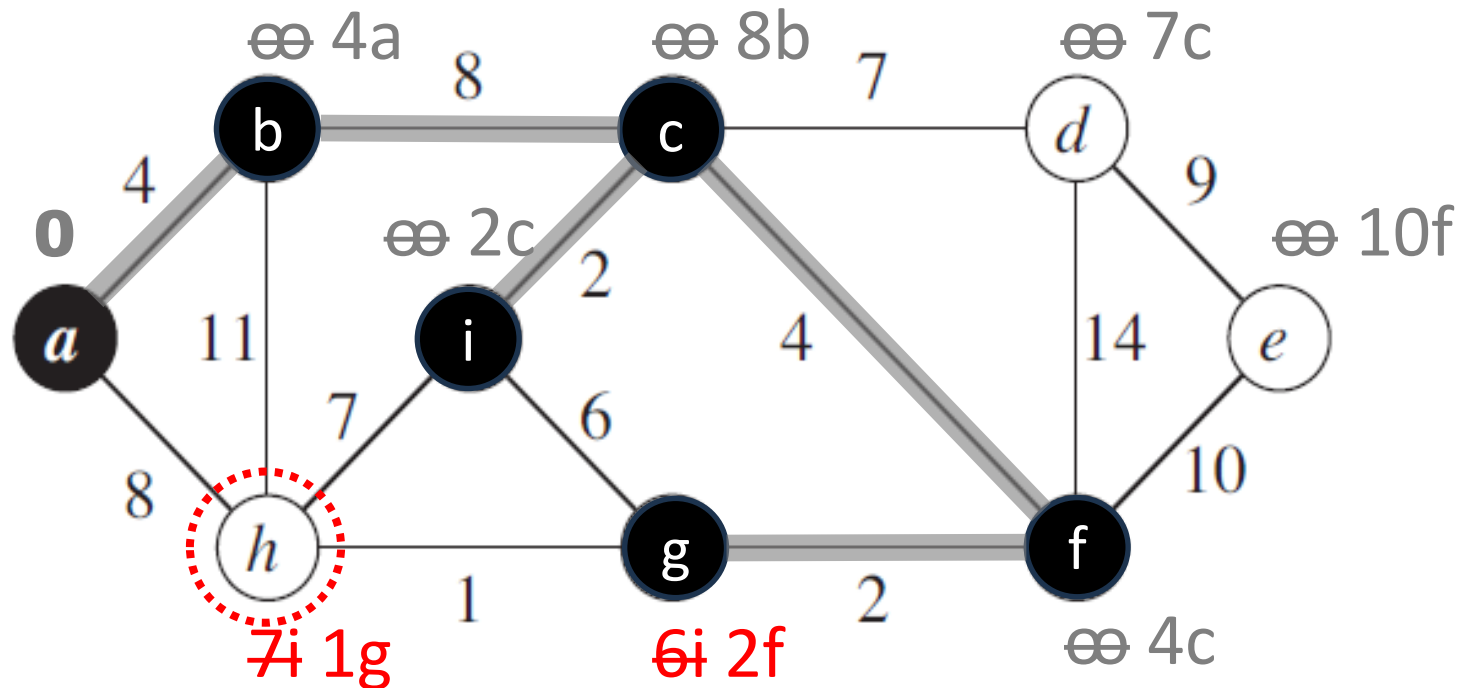
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

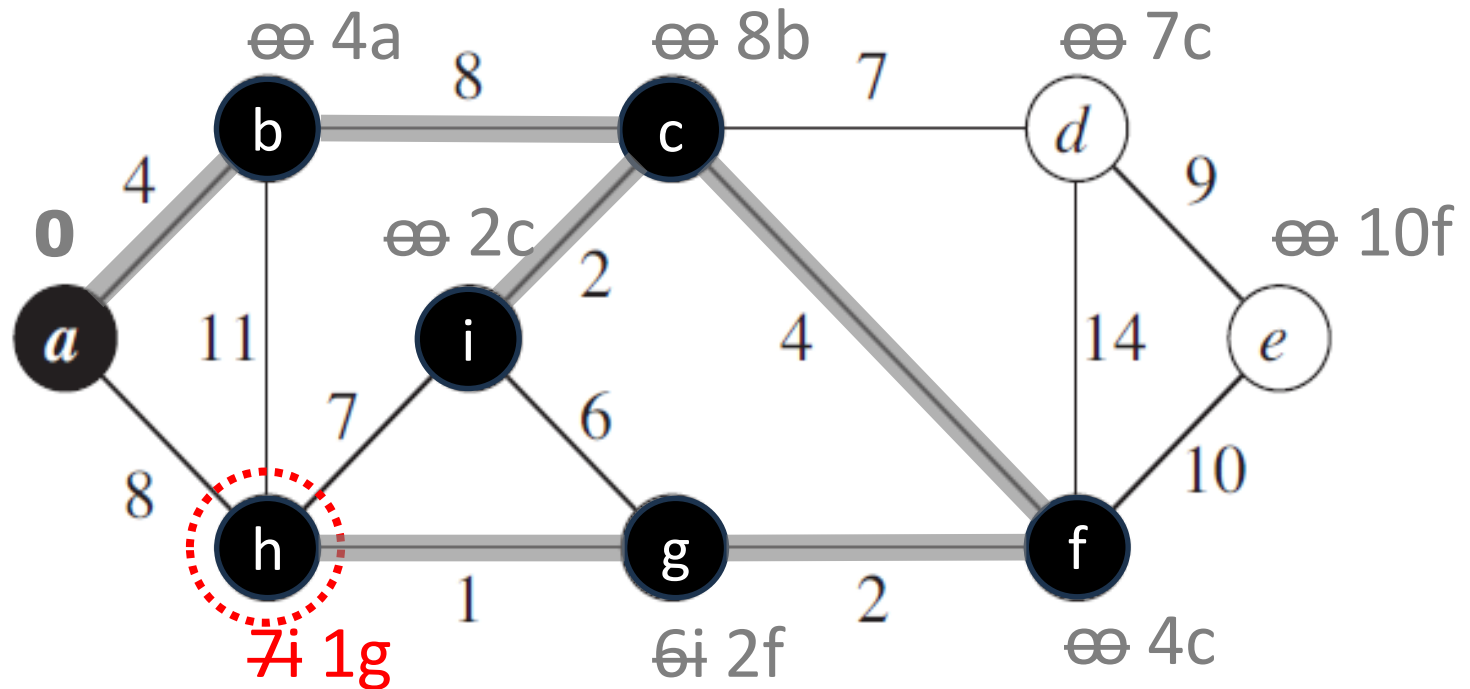
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

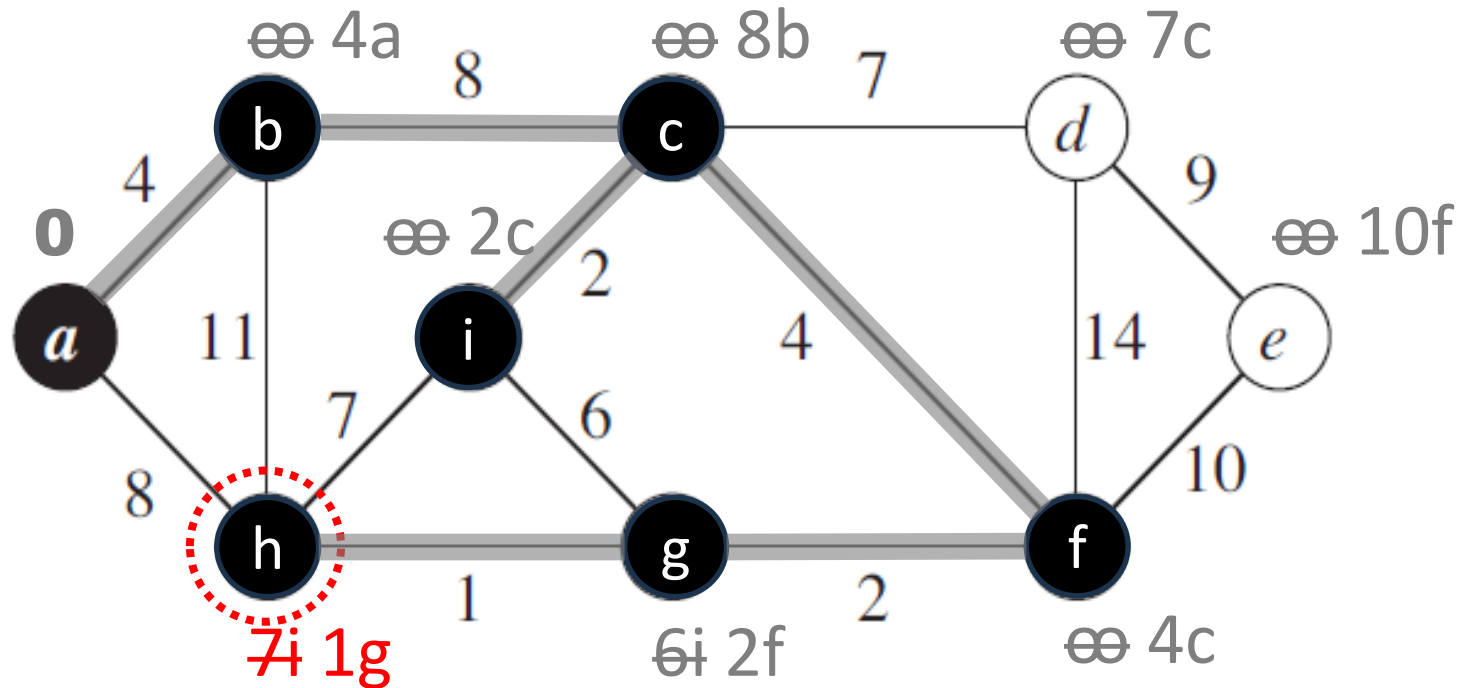
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

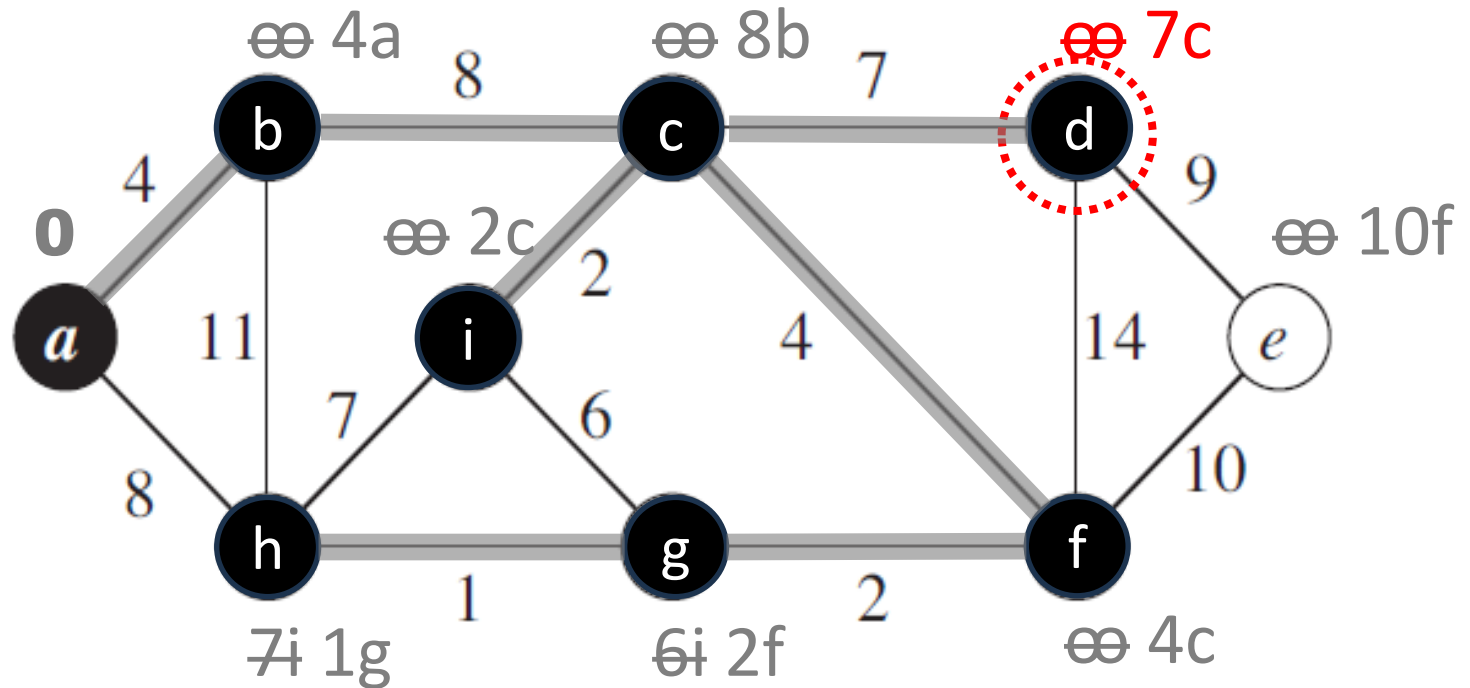
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

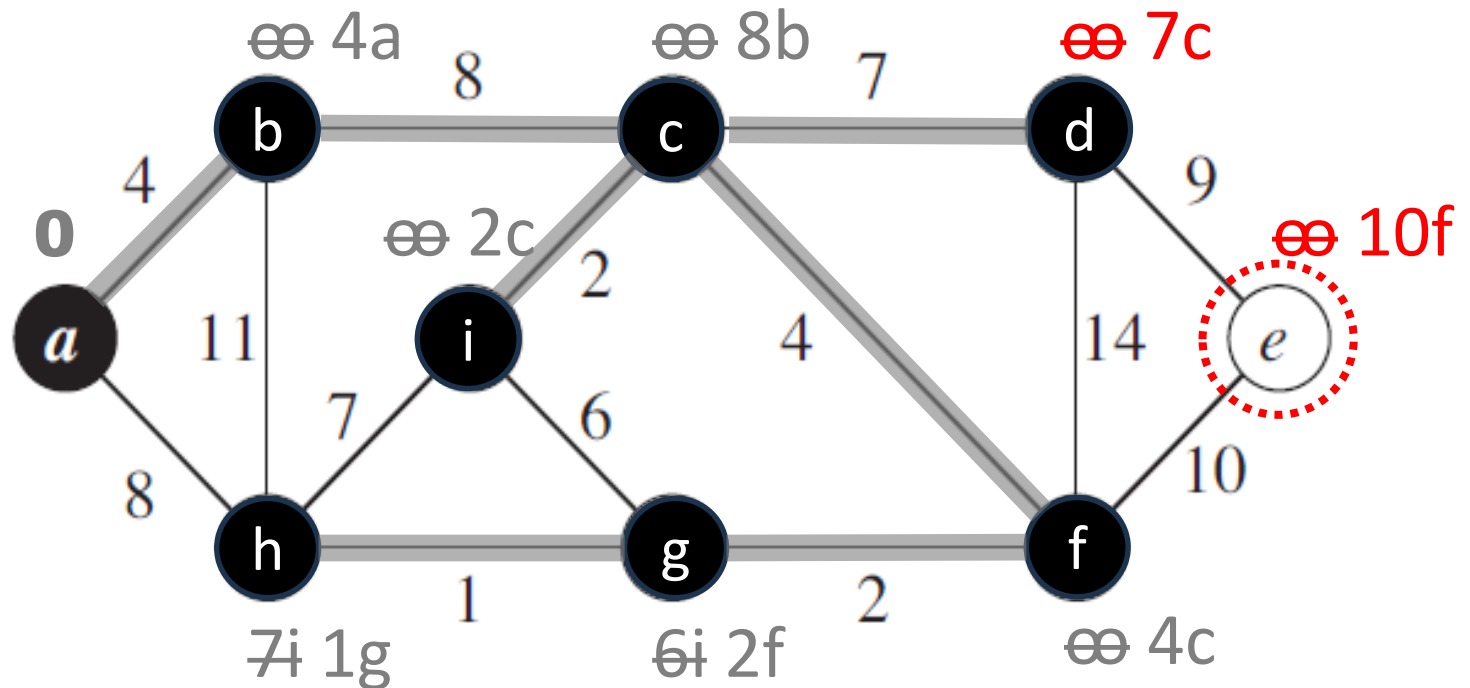
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



# MST-Prim( $G, w, r$ )

...

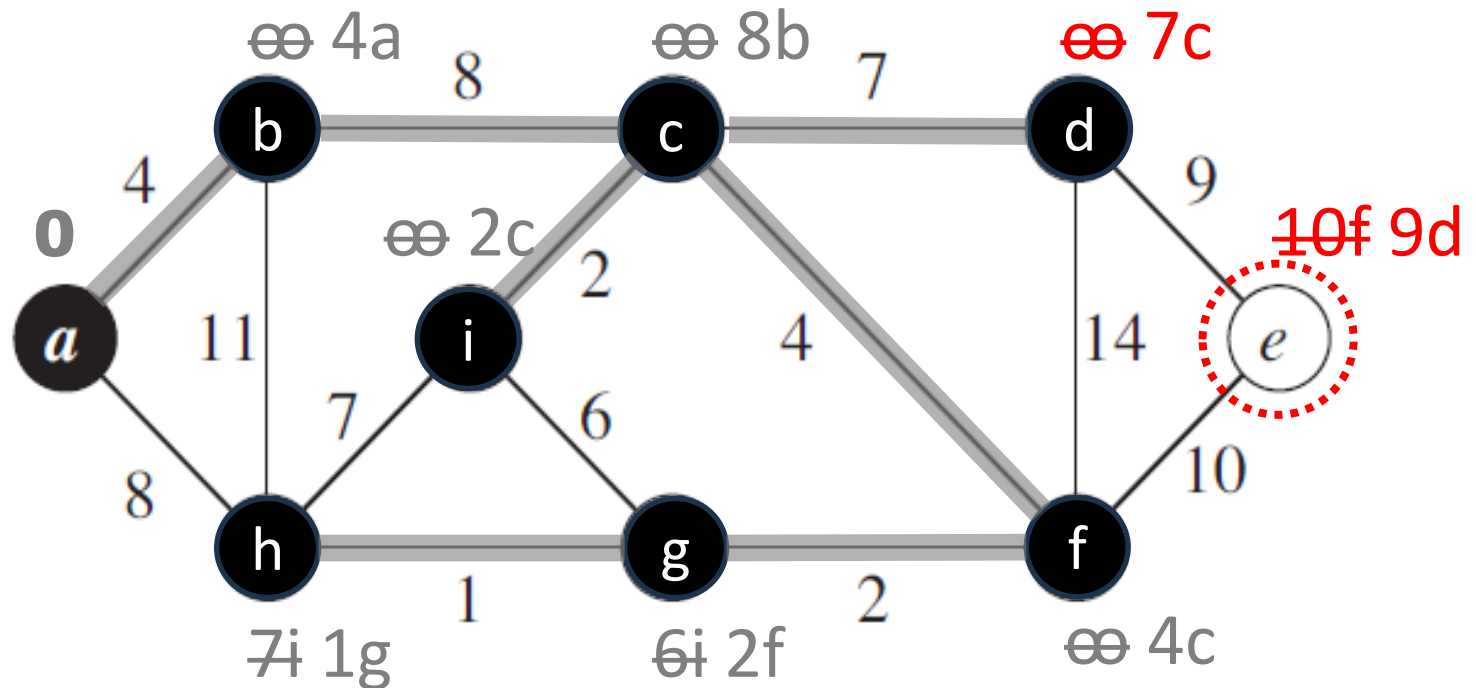
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

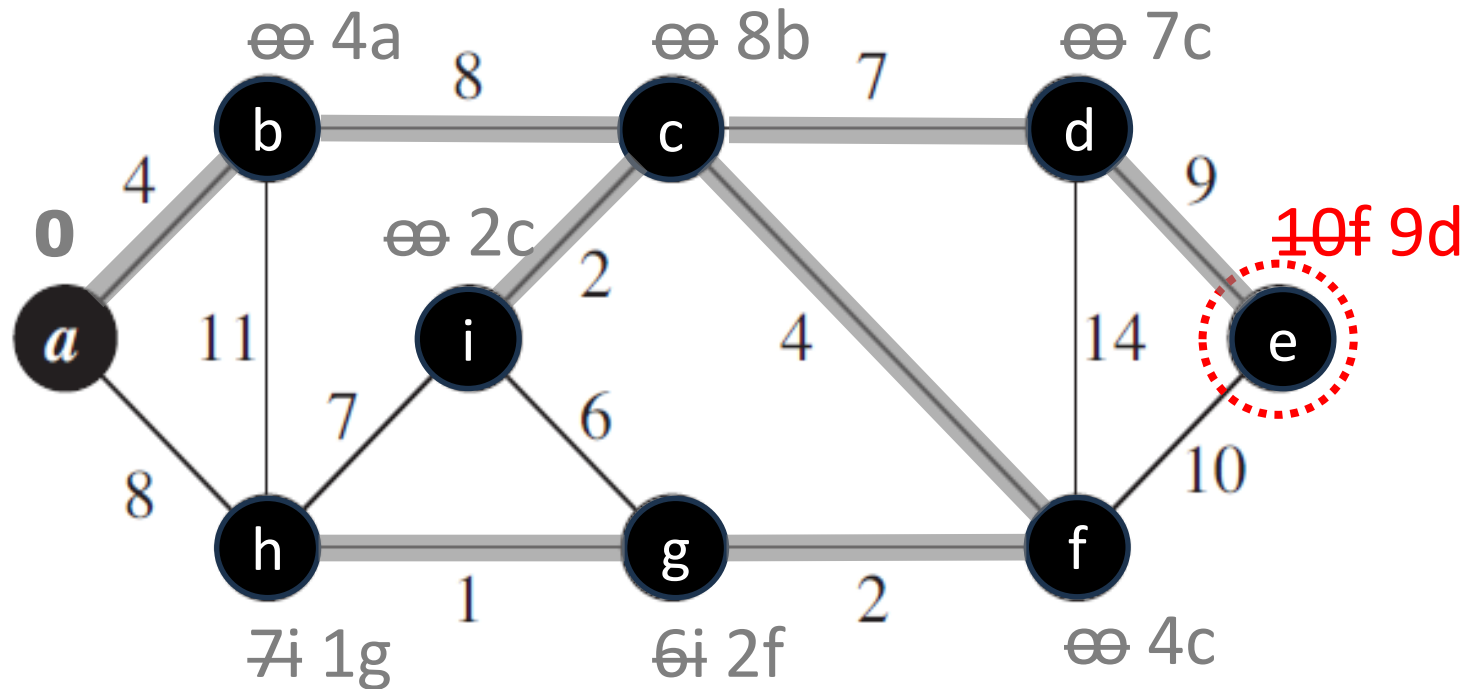
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



## MST-Prim( $G, w, r$ )

...

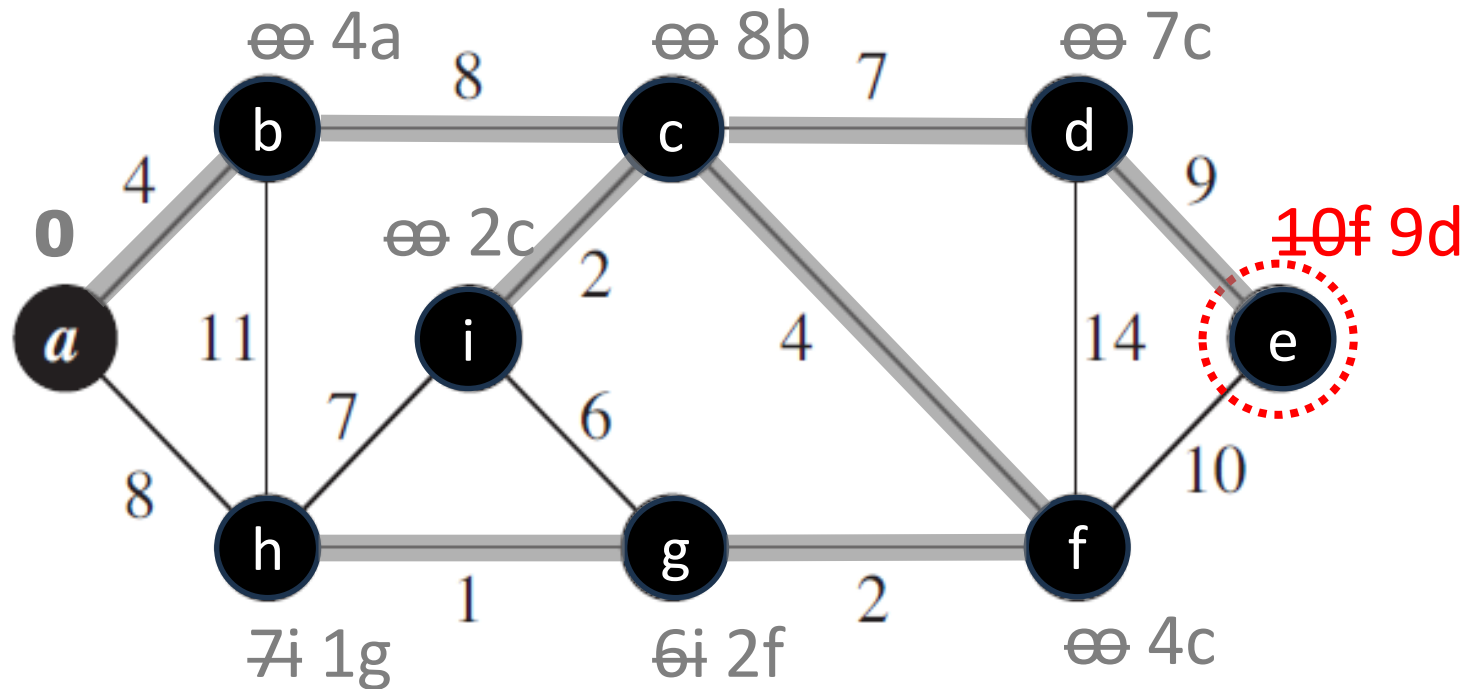
5 enquanto  $Q \neq \text{VAZIO}$  faça

6      $u = \text{ExtraiMinimo}(Q)$

7     para cada  $v$  em  $G.\text{Adj}[u]$  faça

8         se  $v$  está em  $Q$  e  $w(uv) < v.\text{chave}$

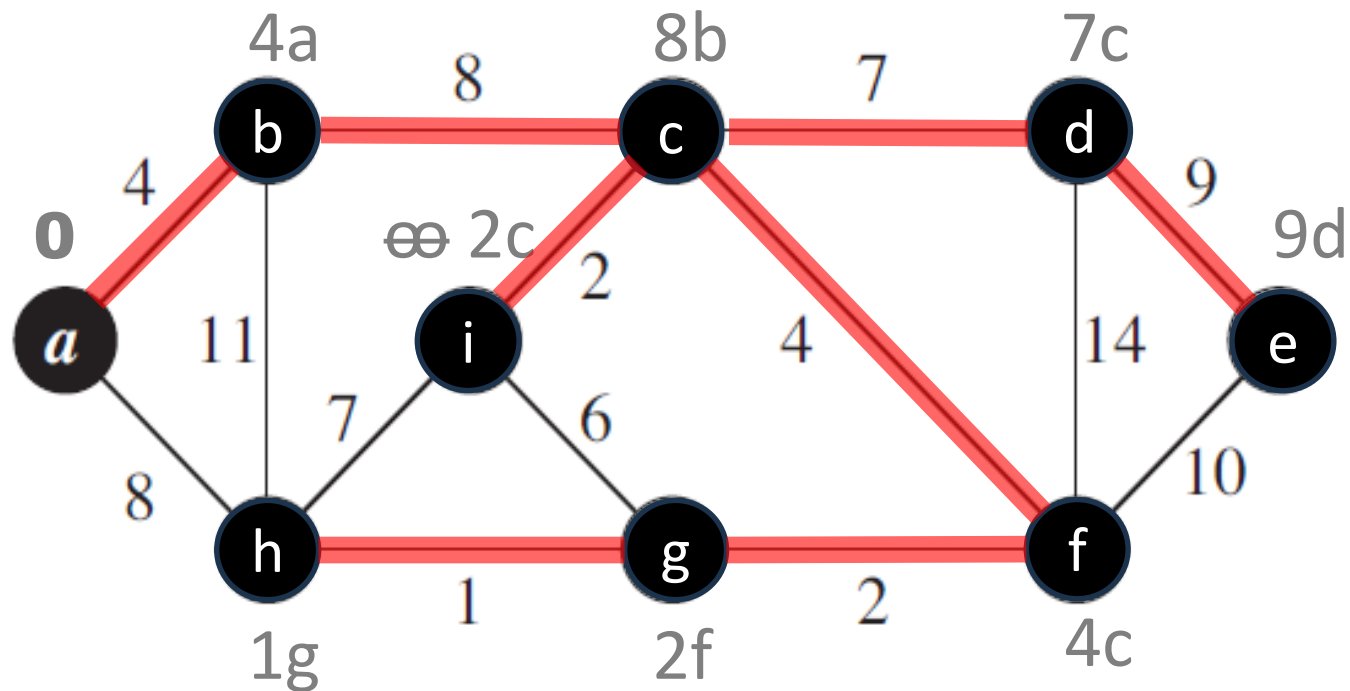
9             então  $v.\text{chave} = w(uv)$ ;  $v.\text{pai} = u$



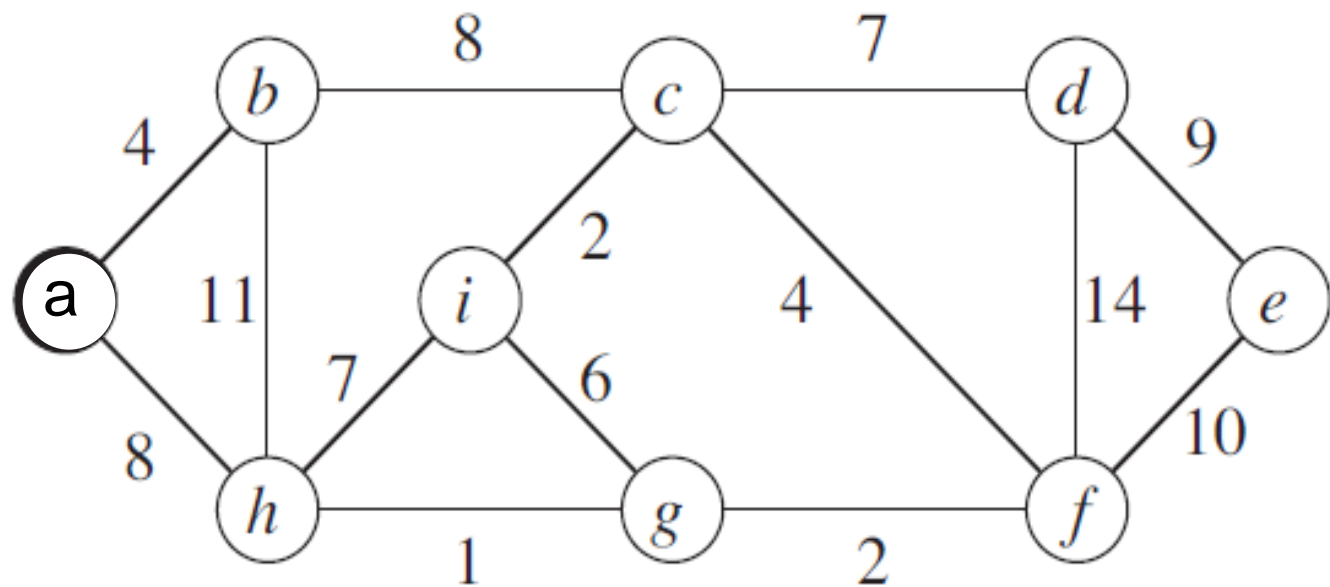


# Exemplo

- **MST:** peso total =  $4+8+7+9+2+4+1+2 = 37$



# Exemplo: MST



**MST-Prim**(G, w, r)

1 para cada vértice u em G.V faça

2     u.**chave** = INFINITO

3     u.**pai** = NIL

4 r.**chave** = 0;    **Q = G.V**

---

5 enquanto **Q != VAZIO** faça

6     u = **ExtraiMinimo**(Q)

7     para cada v em G.Adj[u] faça

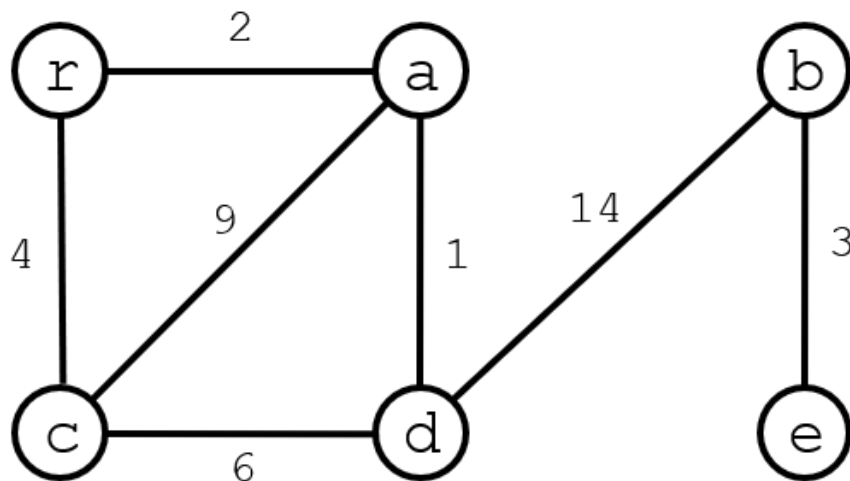
8         se **v está em Q** e  $w(uv) < v.\text{chave}$

9             entao v.**chave** =  $w(uv)$ ; v.**pai** = u

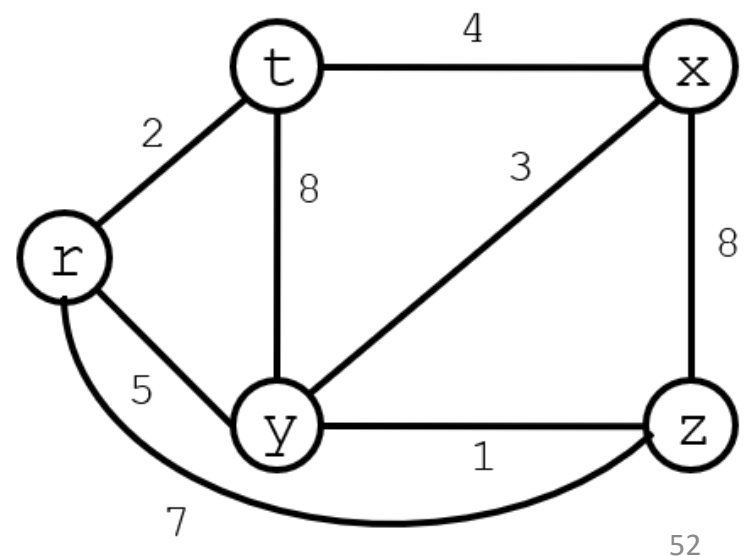
# Exercícios

- Simule o algoritmo de **Prim** para calcular uma **árvore geradora mínima**:

(a)



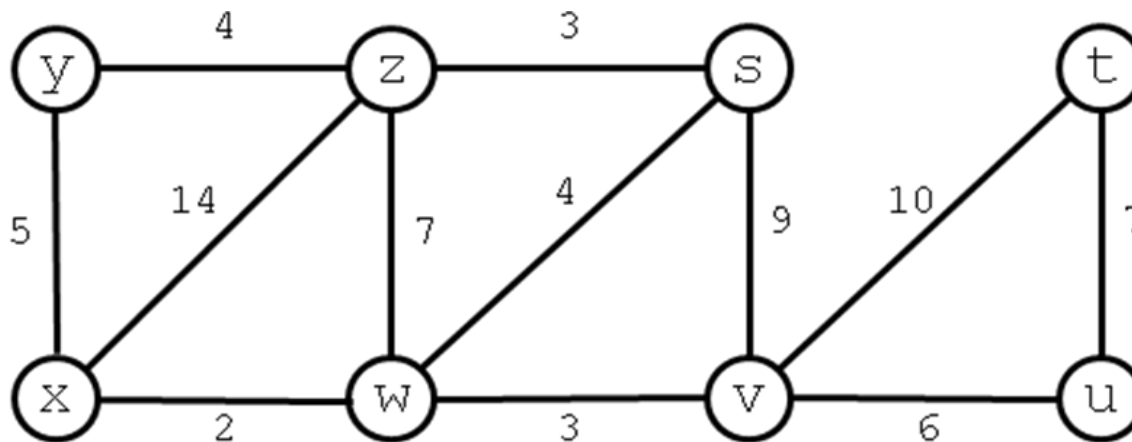
(b)



# Exercícios

- Simule o algoritmo de **Prim** para calcular uma árvore geradora mínima:

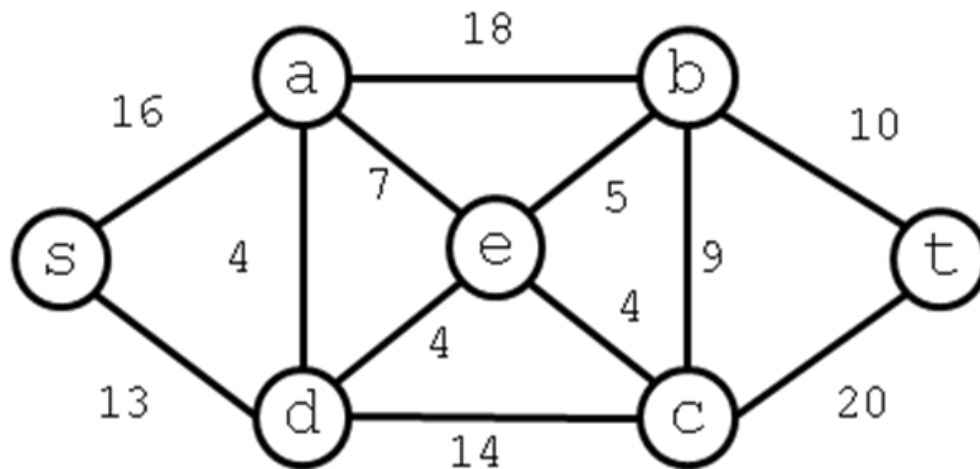
(c)



# Exercícios

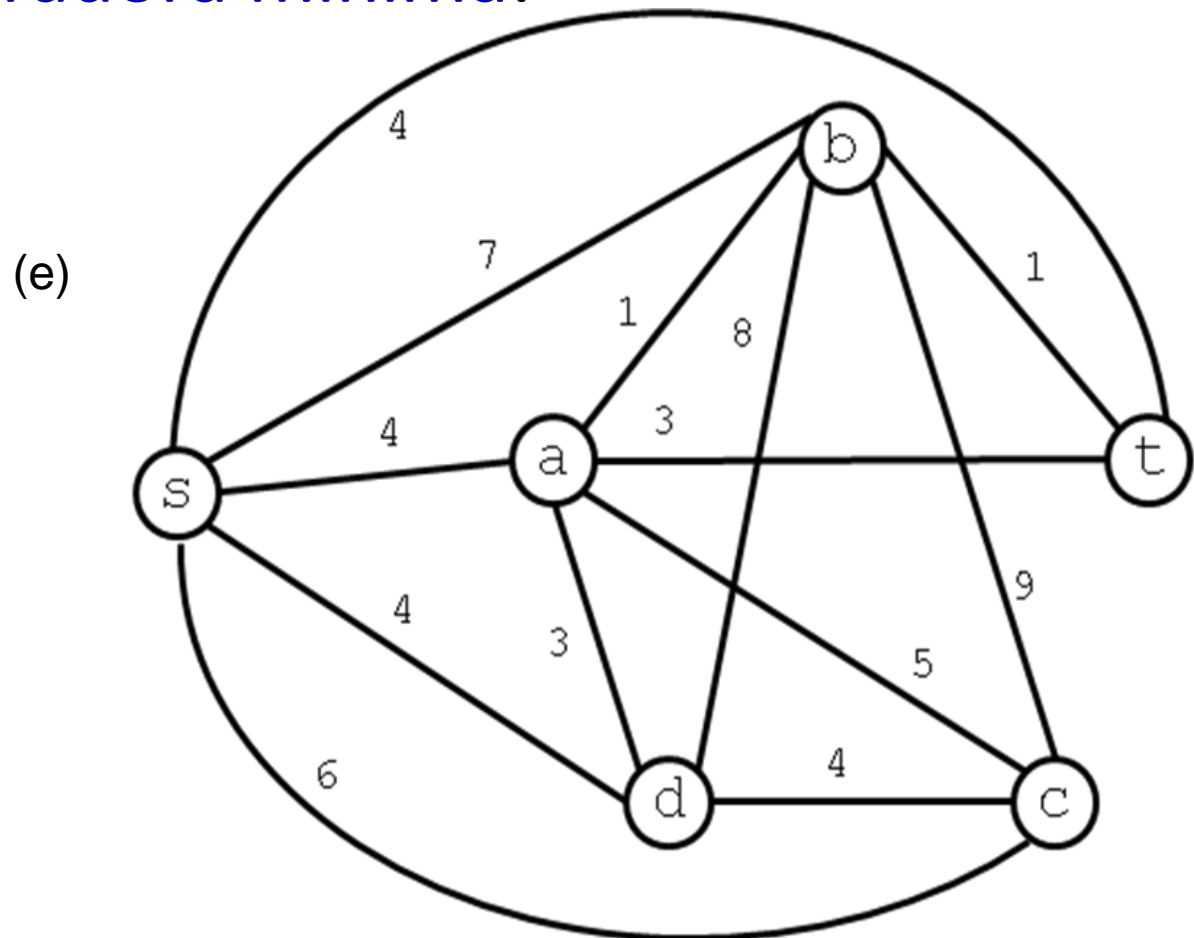
- Simule o algoritmo de **Prim** para calcular uma **árvore geradora mínima**:

(d)



# Exercícios

- Simule o algoritmo de **Prim** para calcular uma árvore geradora mínima:



# Consumo de Tempo?

Consumo  
de tempo:

**MST-Prim**(G, w, r)

1	para cada vértice u em G.V faça	???
2	u.chave = INFINITO	???
3	u.pai = NIL	???
4	r.chave = 0; Q = G.V	???
<hr/>		
5	enquanto Q != VAZIO faça	???
6	u = ExtraiMinimo(Q)	???
7	para cada v em G.Adj[u] faça	???
8	se v está em Q e w(uv) < v.chave	???
9	entao v.chave = w(uv); v.pai = u	???
10	diminuiChave(Q, v)	???

**Total:**

**T(n, m) = ???**



# Consumo de Tempo?

Consumo  
de tempo:

**MST-Prim**(G, w, r)

1	para cada vértice u em G.V faça	$O(n)$
2	u.chave = INFINITO	$O(n)$
3	u.pai = NIL	$O(n)$
4	r.chave = 0; <b>Q = G.V</b>	$O(1) + O(n)$
<hr/>		
5	enquanto <b>Q != VAZIO</b> faça	$O(n)$
6	u = <b>ExtraiMinimo</b> (Q)	$O(n) * O(\log n)$
7	para cada v em G.Adj[u] faça	$O(m)$
8	se <b>v está em Q</b> e $w(uv) < v.chave$	$O(m)$
9	entao v.chave = $w(uv)$ ; v.pai = u	$O(m)$
10	<b>diminuiChave</b> (Q, v)	$O(m) * O(\log n)$

**Total:**

**$T(n, m) = O((n+m) * \log n)$**

# Exercício Programa

- 12-mst-prim.py