

Introdução à Teoria dos Grafos

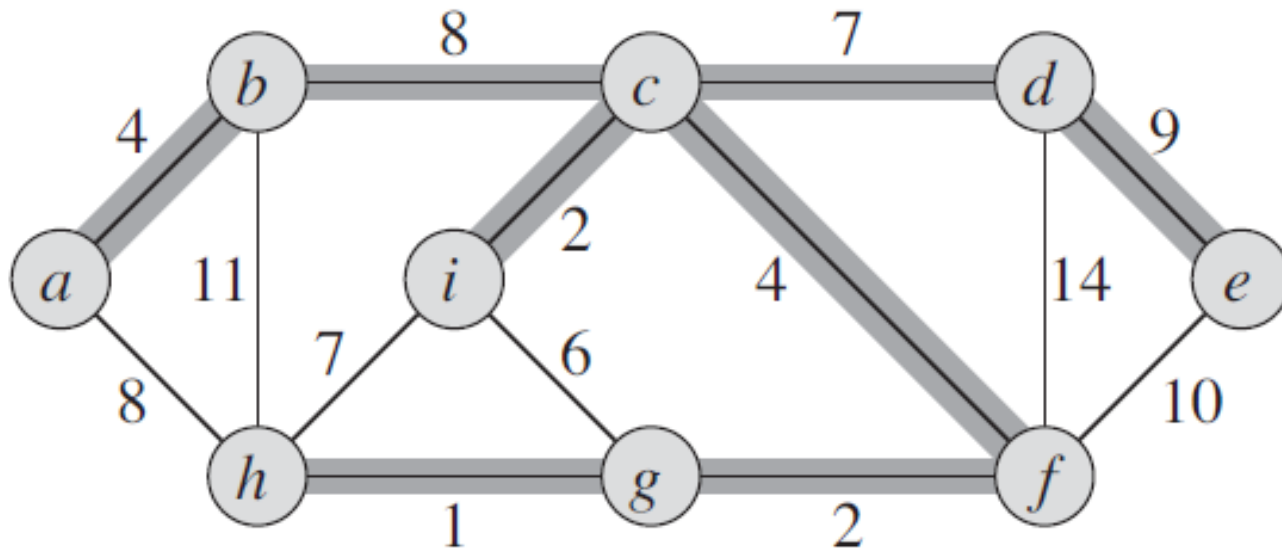
Prof. Alexandre Noma

Aula passada

- **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**

Problema: **MST**

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



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

Hoje

- Kruskal(G, w)
 - Entrada: um grafo conexo **G** ,
ponderado com pesos **w** .
Saída: **árvore geradora mínima.**
- Operações com **conjuntos disjuntos**
 - **makeSet**(x)
 - **findSet**(x)
 - **Union**(x, y)

Conjuntos disjuntos?

- Coleção $S = \{S_1, S_1, \dots, S_k\}$ de conjuntos disjuntos.

Conjuntos disjuntos?

- Coleção $S = \{S_1, S_1, \dots, S_k\}$ de conjuntos disjuntos.
- Cada conjunto é identificado por um representante.
- Ex.

| | | | | |
|---------------------------|---------------------|------------------|---------------------|------------------|
| S_1 | S_2 | S_3 | S_4 | S_5 |
| $\{\mathbf{a}, b, c, d\}$ | $\{e, \mathbf{g}\}$ | $\{\mathbf{f}\}$ | $\{h, \mathbf{i}\}$ | $\{\mathbf{j}\}$ |

Conjuntos disjuntos

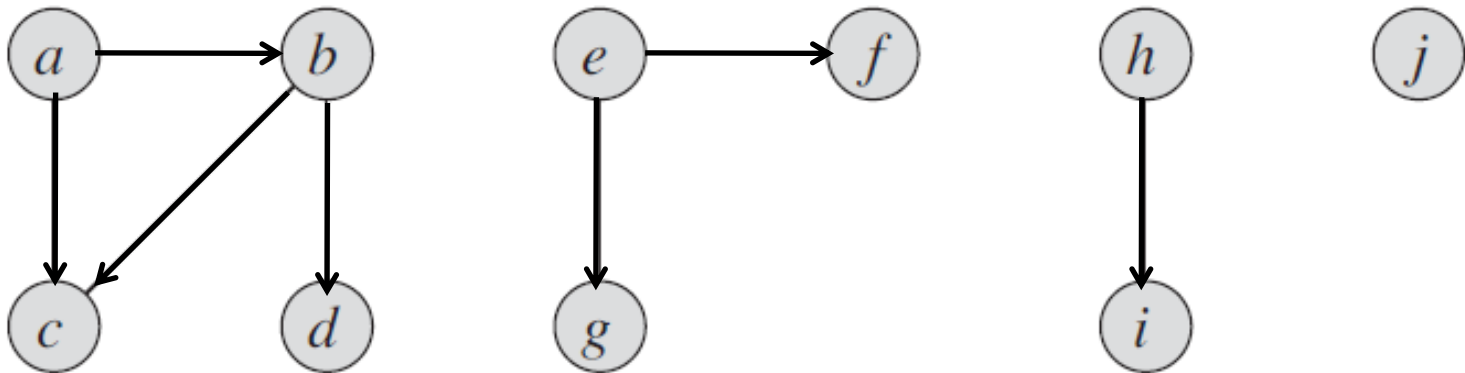
- Coleção $S = \{S_1, S_1, \dots, S_k\}$ de conjuntos disjuntos.
- Cada conjunto é identificado por um representante.
- Ex.

| | | | | |
|----------------------------|---------------------|------------------|---------------------|------------------|
| S_1 | S_2 | S_3 | S_4 | S_5 |
| $-\{\mathbf{a}, b, c, d\}$ | $\{e, \mathbf{g}\}$ | $\{\mathbf{f}\}$ | $\{h, \mathbf{i}\}$ | $\{\mathbf{j}\}$ |
- Operações
 - **FindSet**(x): Devolve o representante de S_x .
 - **MakeSet**(x): Cria novo conjunto com um único elemento x.
 - **Union**(x, y): Une dois conjuntos S_x e S_y .

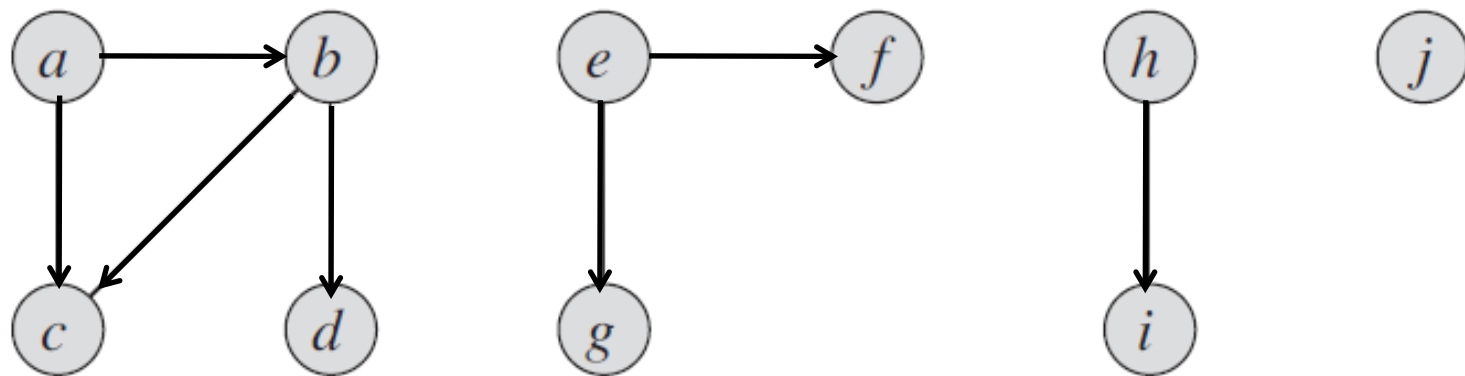
Exemplo

ConnectedComponents (G)

- 1 para cada vértice u em $G.V$ faça
- 2 **MakeSet** (u)
- 3 para cada uv em $G.E$ faça
- 4 se **FindSet** (u) \neq **FindSet** (v)
- 5 entao **Union** (u, v)



| Edge processed | Collection of disjoint sets | | | | | | | | | |
|----------------|-----------------------------|-----------|---------|---------|-------------|---------|---------|-----------|---------|---------|
| initial sets | $\{a\}$ | $\{b\}$ | $\{c\}$ | $\{d\}$ | $\{e\}$ | $\{f\}$ | $\{g\}$ | $\{h\}$ | $\{i\}$ | $\{j\}$ |
| (b,d) | $\{a\}$ | $\{b,d\}$ | $\{c\}$ | | $\{e\}$ | $\{f\}$ | $\{g\}$ | $\{h\}$ | $\{i\}$ | $\{j\}$ |
| (e,g) | $\{a\}$ | $\{b,d\}$ | $\{c\}$ | | $\{e,g\}$ | $\{f\}$ | | $\{h\}$ | $\{i\}$ | $\{j\}$ |
| (a,c) | $\{a,c\}$ | $\{b,d\}$ | | | $\{e,g\}$ | $\{f\}$ | | $\{h\}$ | $\{i\}$ | $\{j\}$ |
| (h,i) | $\{a,c\}$ | $\{b,d\}$ | | | $\{e,g\}$ | $\{f\}$ | | $\{h,i\}$ | | $\{j\}$ |
| (a,b) | $\{a,b,c,d\}$ | | | | $\{e,g\}$ | $\{f\}$ | | $\{h,i\}$ | | $\{j\}$ |
| (e,f) | $\{a,b,c,d\}$ | | | | $\{e,f,g\}$ | | | $\{h,i\}$ | | $\{j\}$ |
| (b,c) | $\{a,b,c,d\}$ | | | | $\{e,f,g\}$ | | | $\{h,i\}$ | | $\{j\}$ |



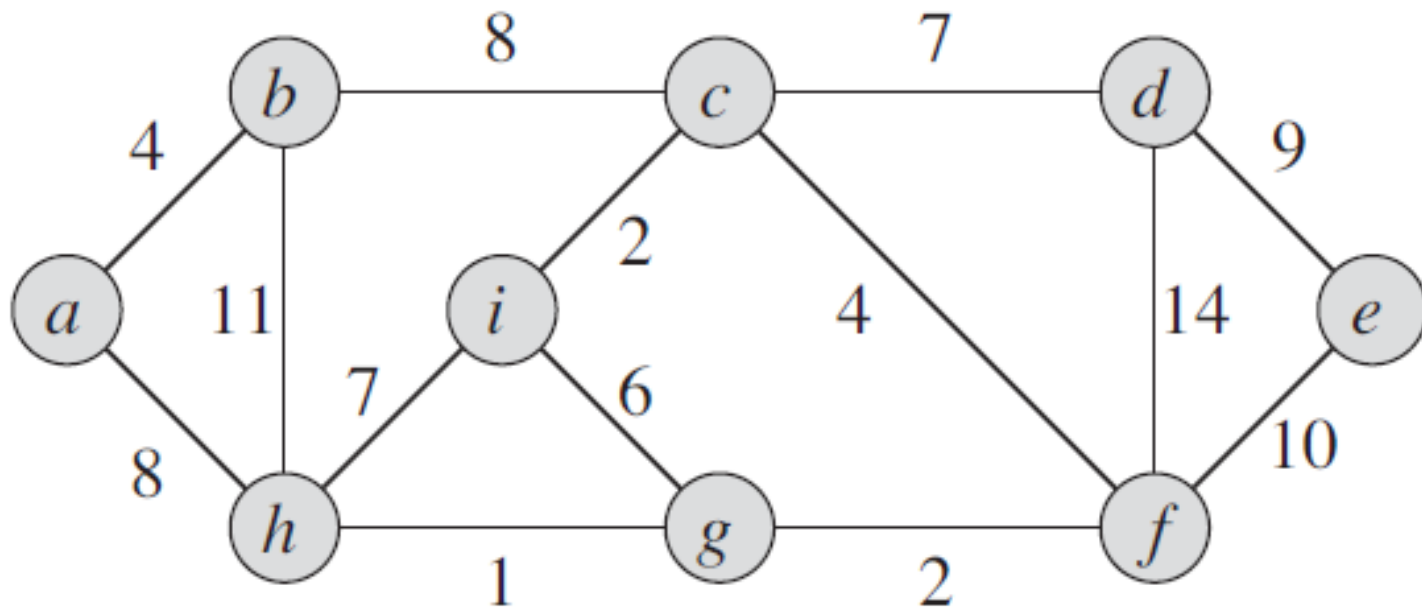
Exercício Programa

- 13-conjuntosDisjuntos.py

MST

- Kruskal(G, w)
 - Entrada: um grafo conexo **G** ,
ponderado com pesos **w** .
Saída: **árvore geradora mínima**.
- Operações com **conjuntos disjuntos**
 - **makeSet**(x)
 - **findSet**(x)
 - **Union**(x, y)

Exemplo



MST-Kruskal (G, w)

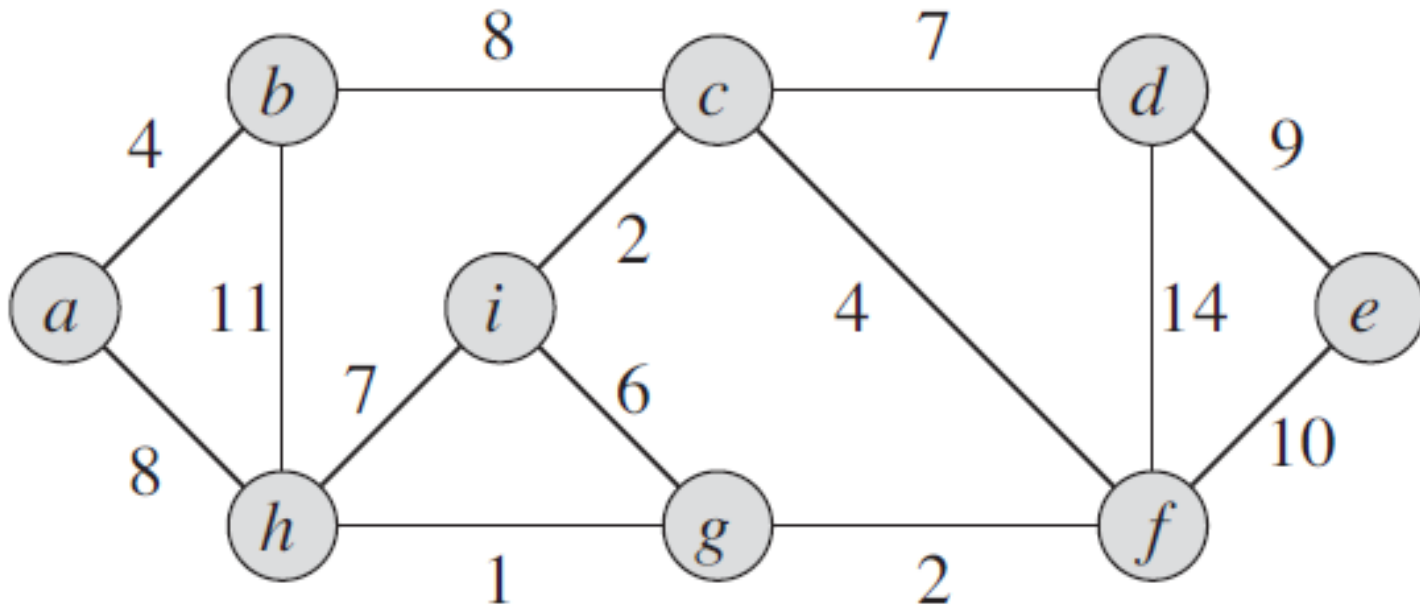
```
1   $T = \emptyset$ 
2  para cada vértice  $u$  em  $G.V$  faça
3      MakeSet( $u$ )
4  ordenar arestas  $G.E$  por peso (crescente)
5  para cada aresta  $uv$  em  $G.E$  (ordenada) faça
6      se FindSet( $u$ )  $\neq$  FindSet( $v$ )
7          então Union( $u, v$ )
8           $T = T \cup \{uv\}$ 
9  devolva  $T$ 
```

MST-Kruskal (G, w)

- 1 $T = \emptyset$
- 2 para cada vértice u em $G.V$ faça
- 3 **MakeSet** (u)
- 4 ordenar arestas $G.E$ por peso (crescente)

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



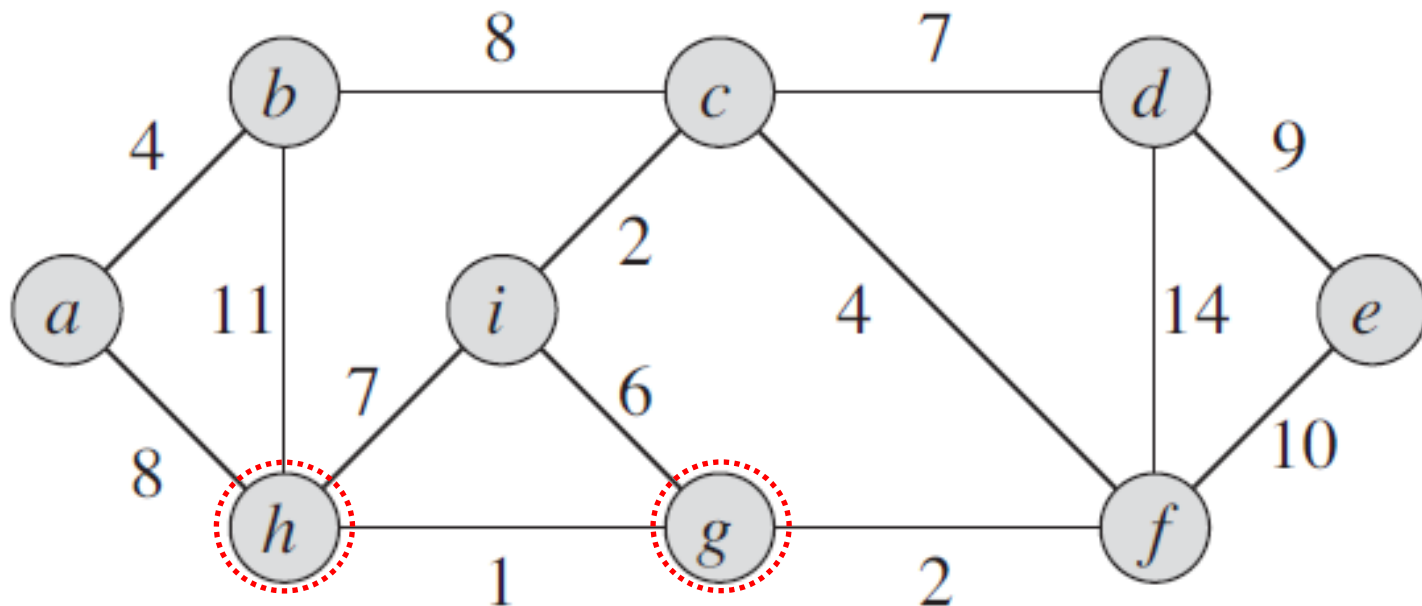
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



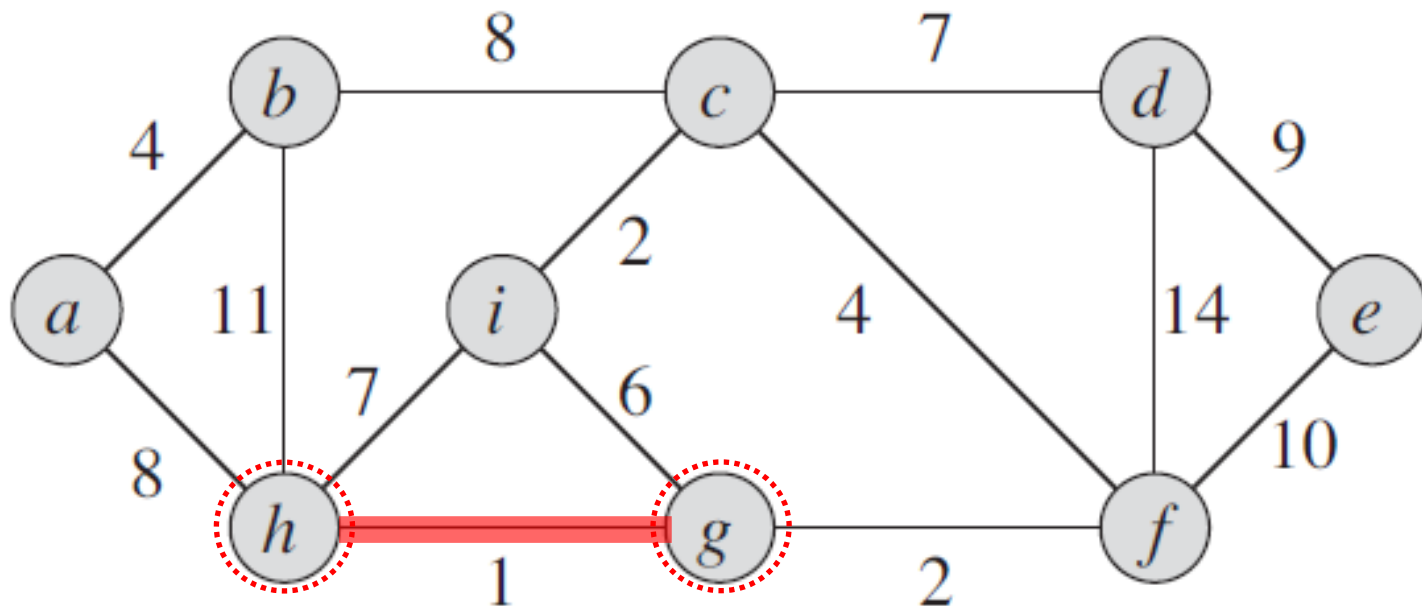
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



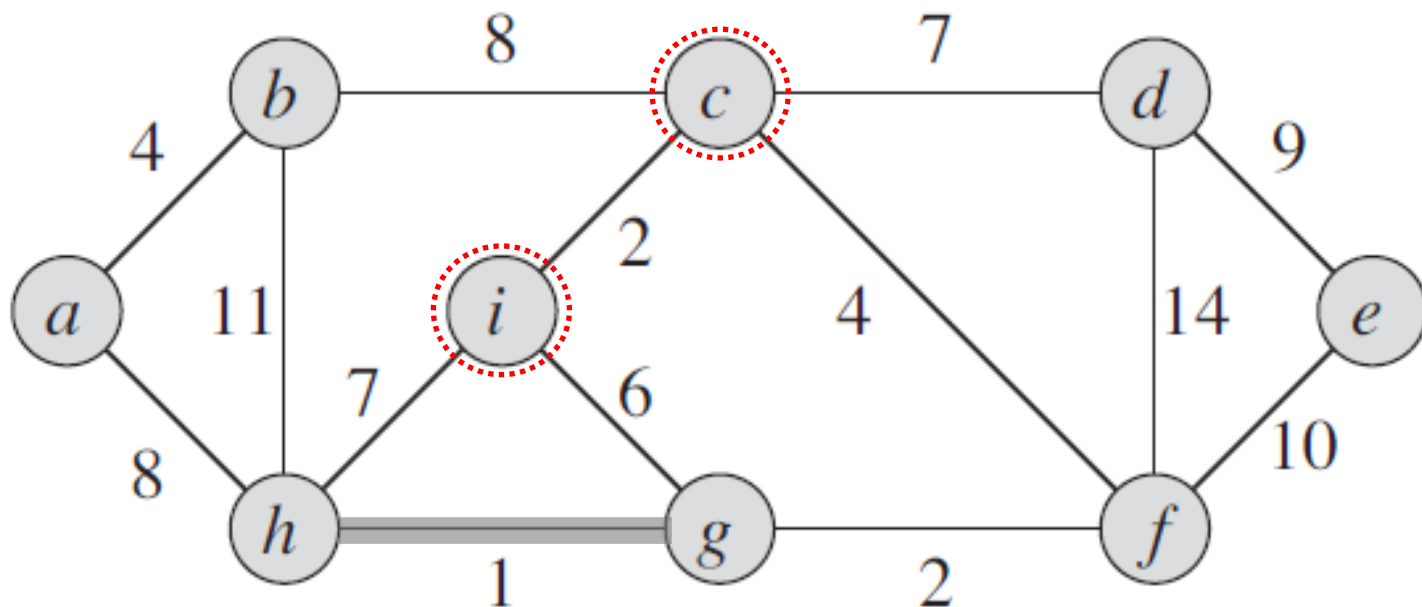

```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



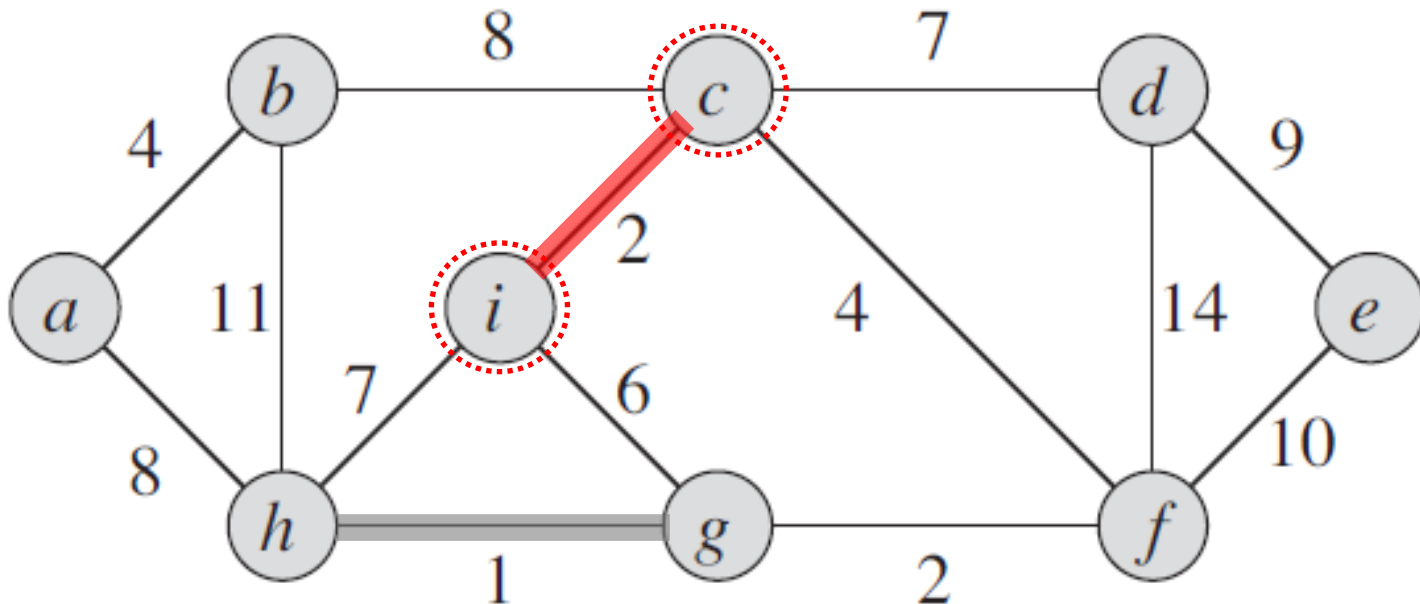
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



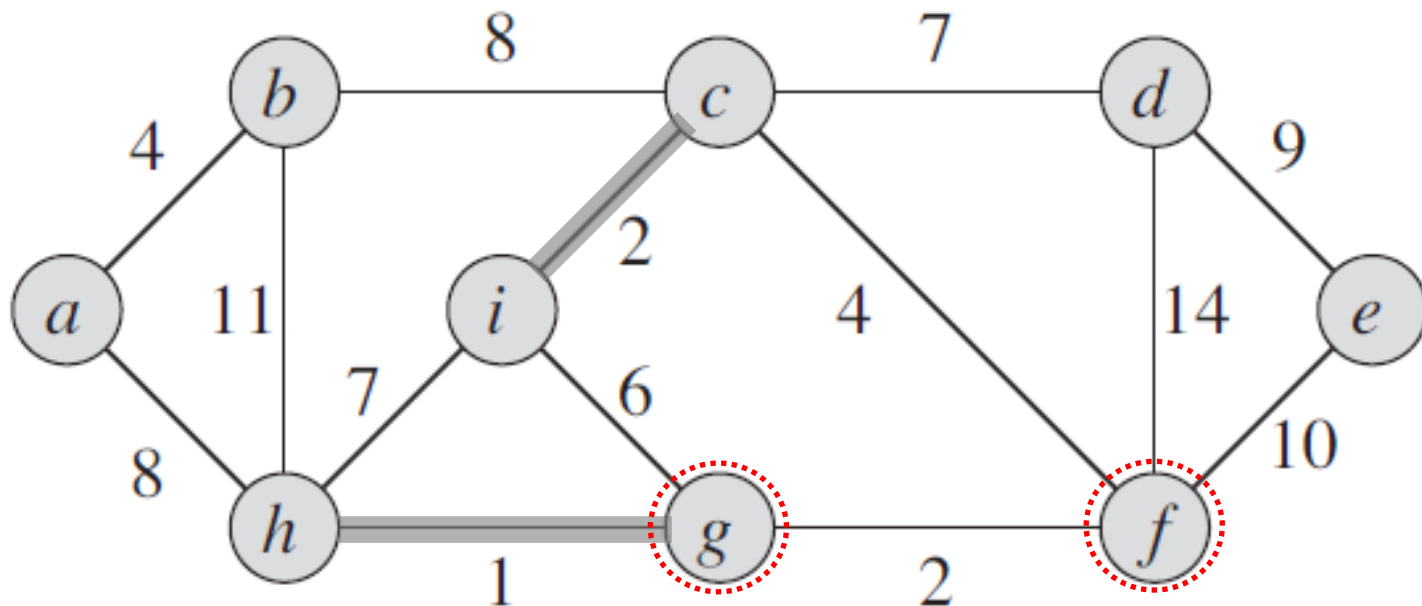
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



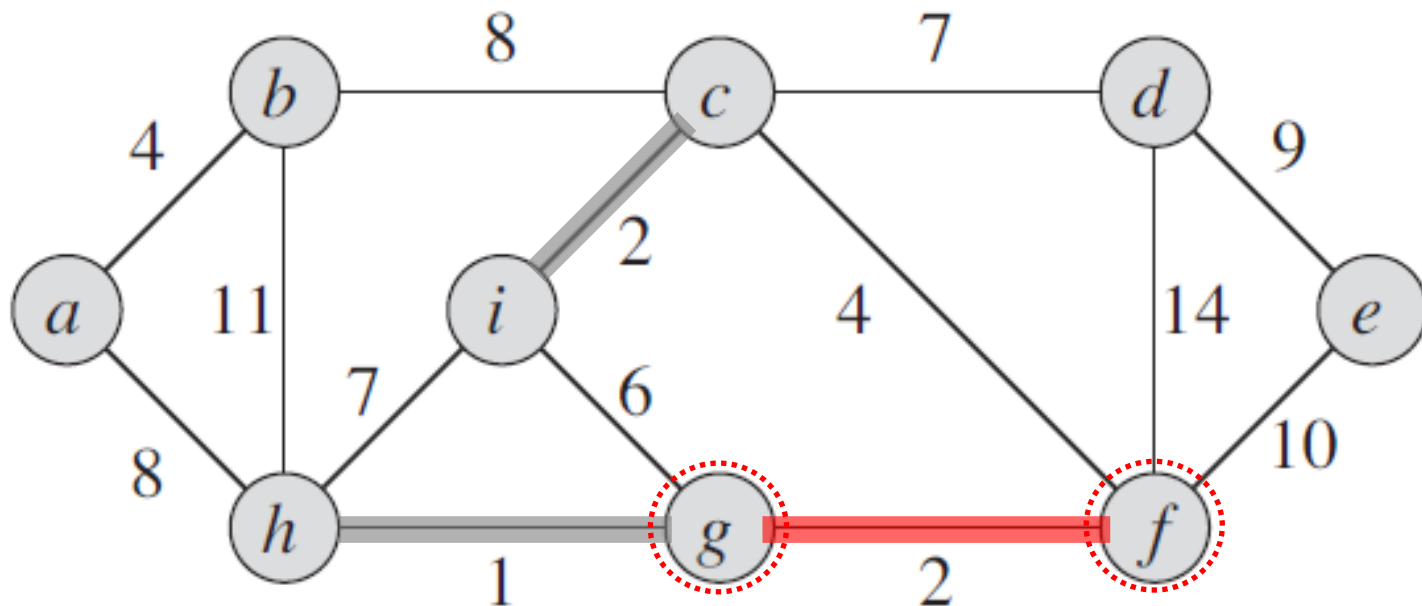
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



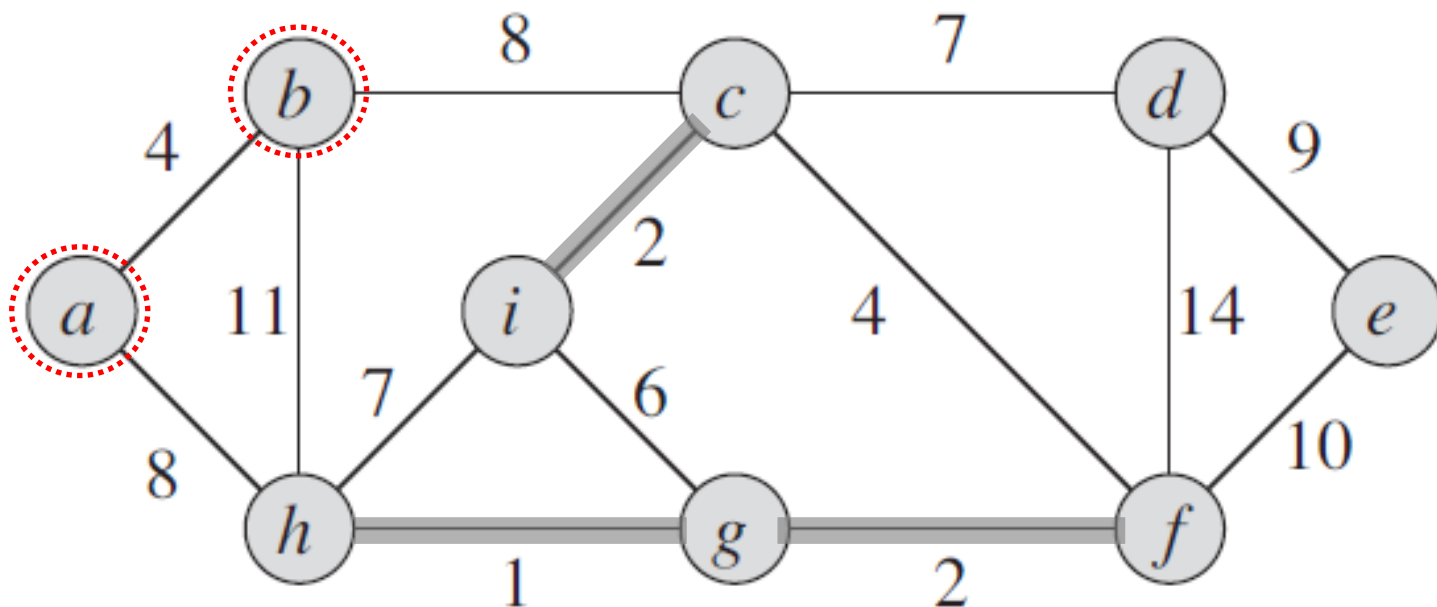
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



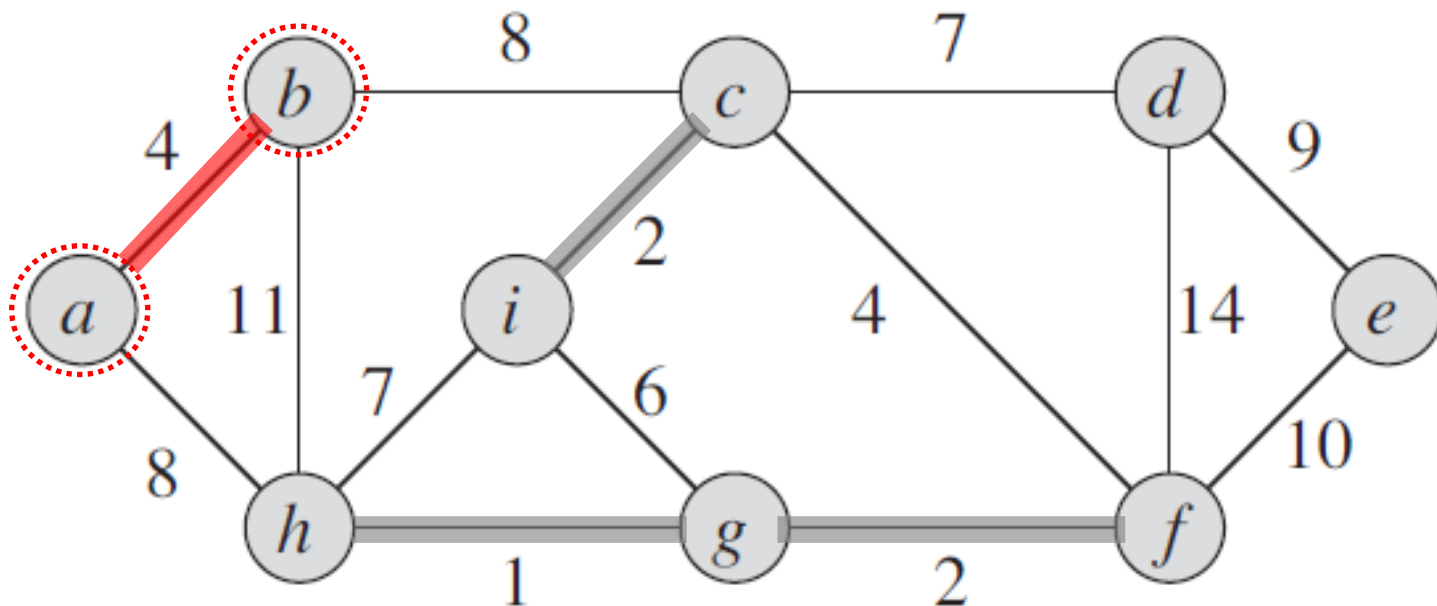
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



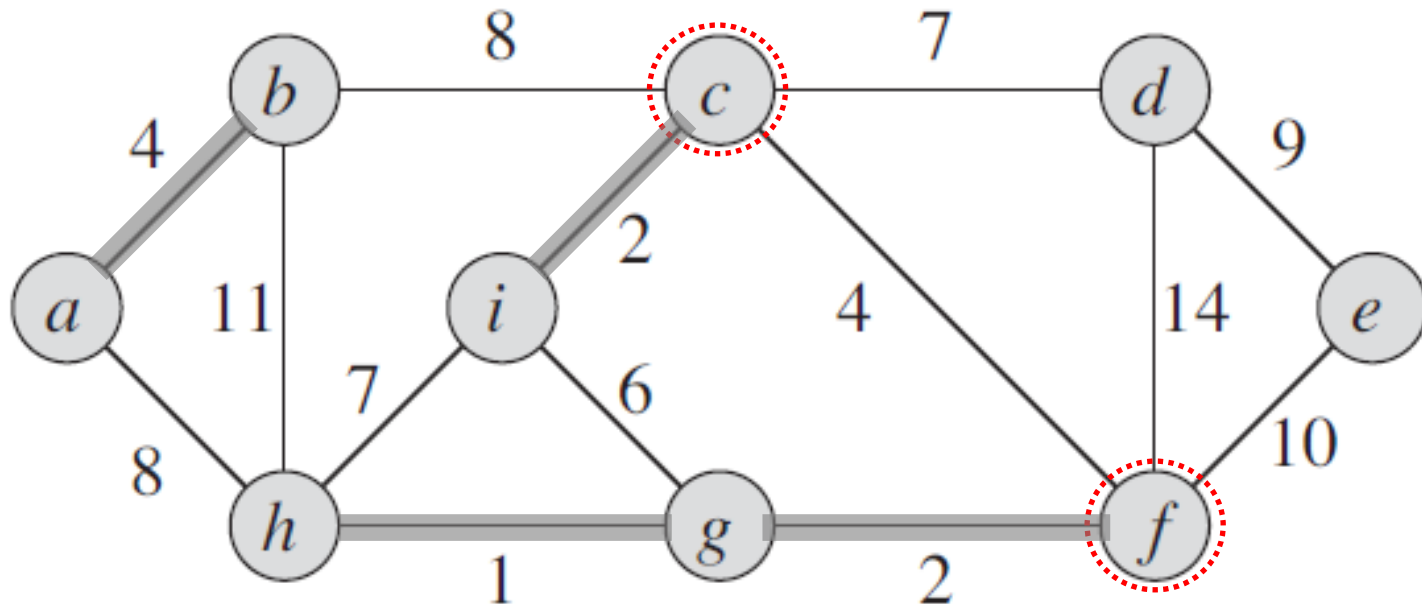
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



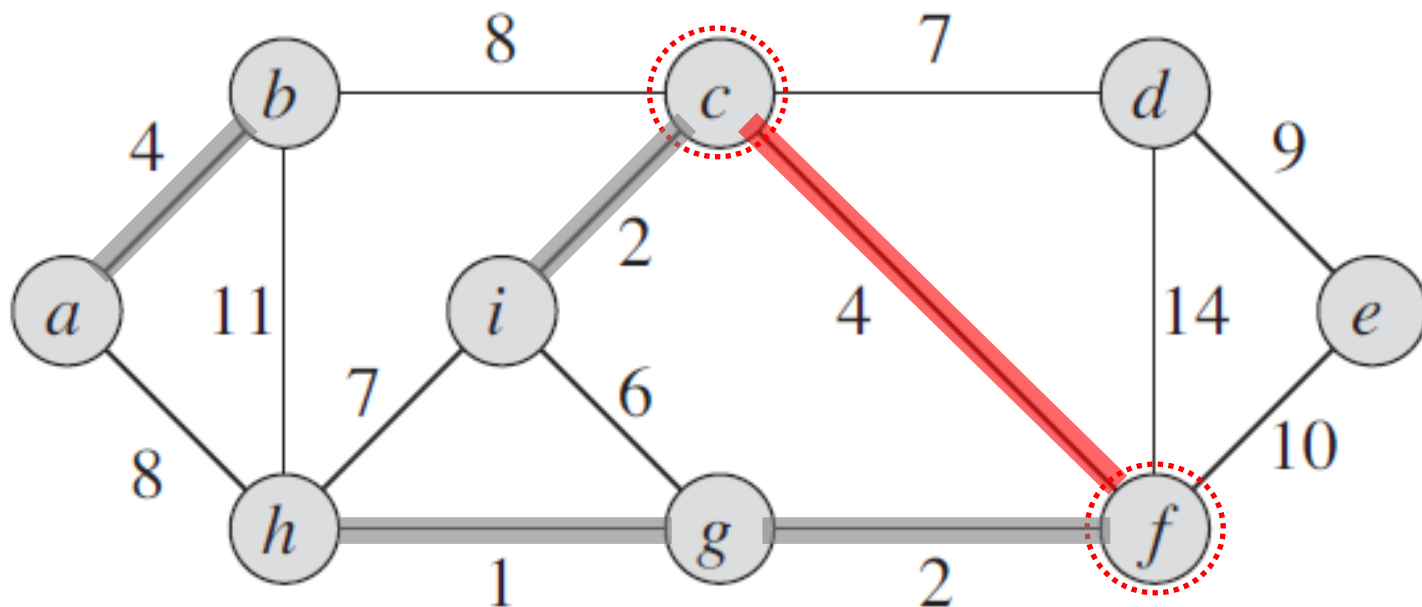
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



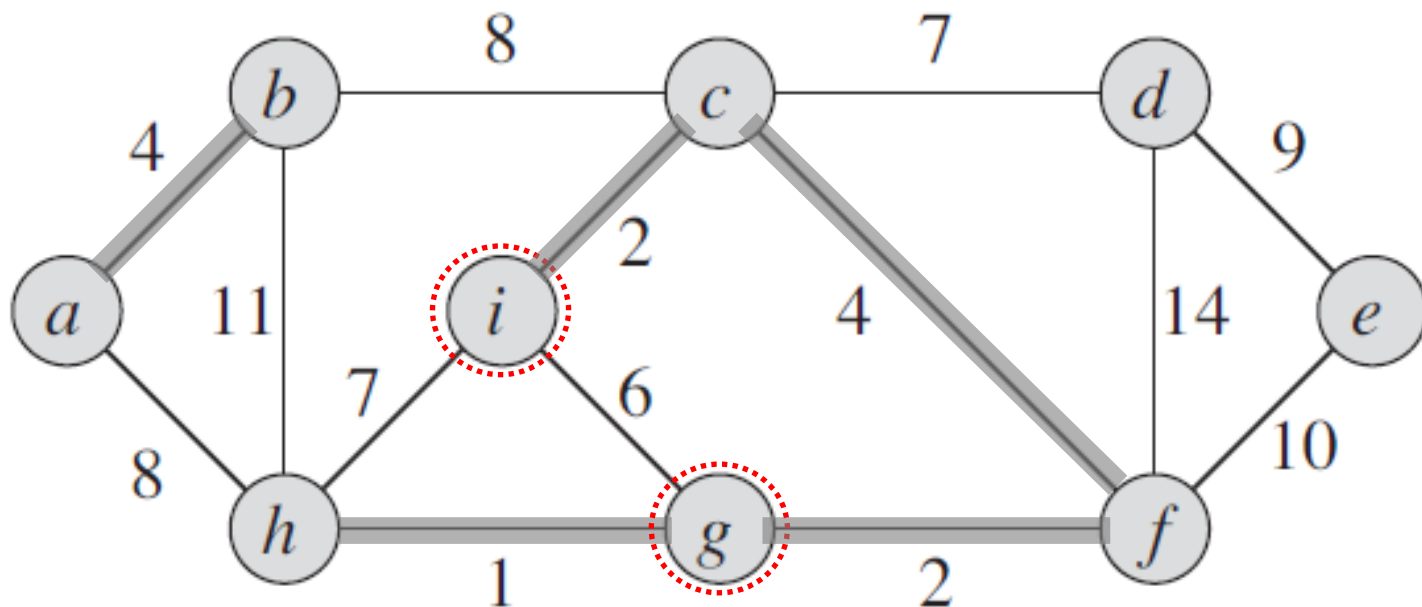

```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



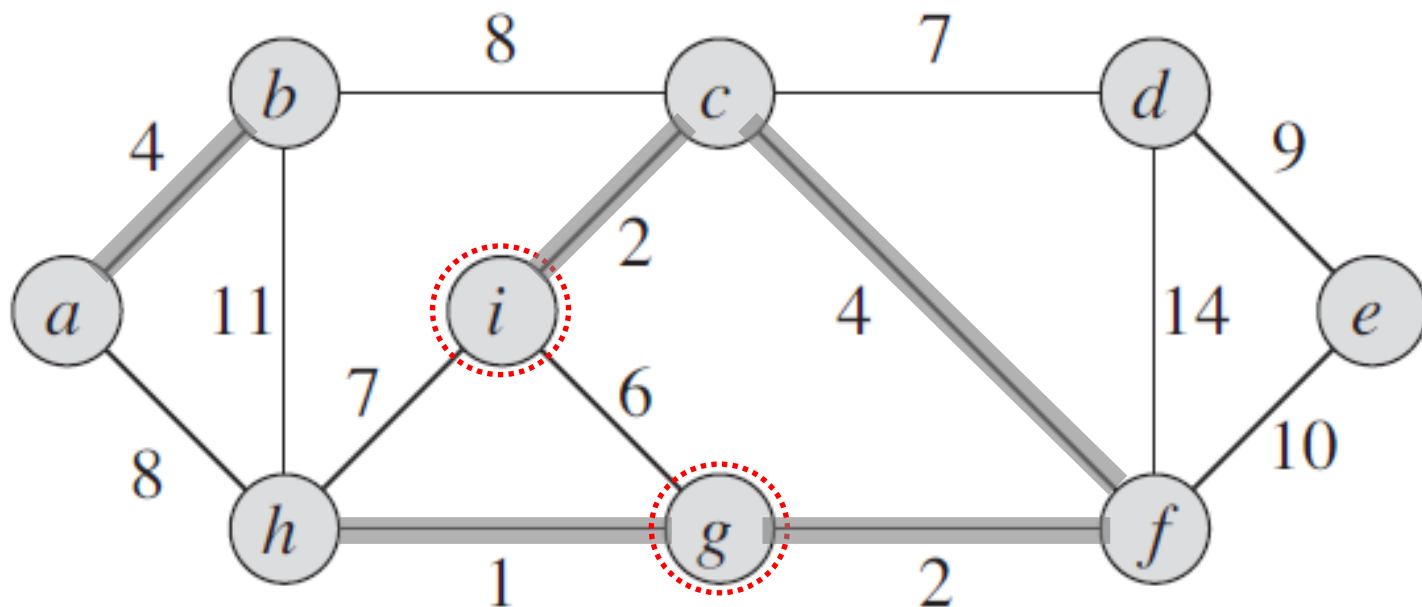
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



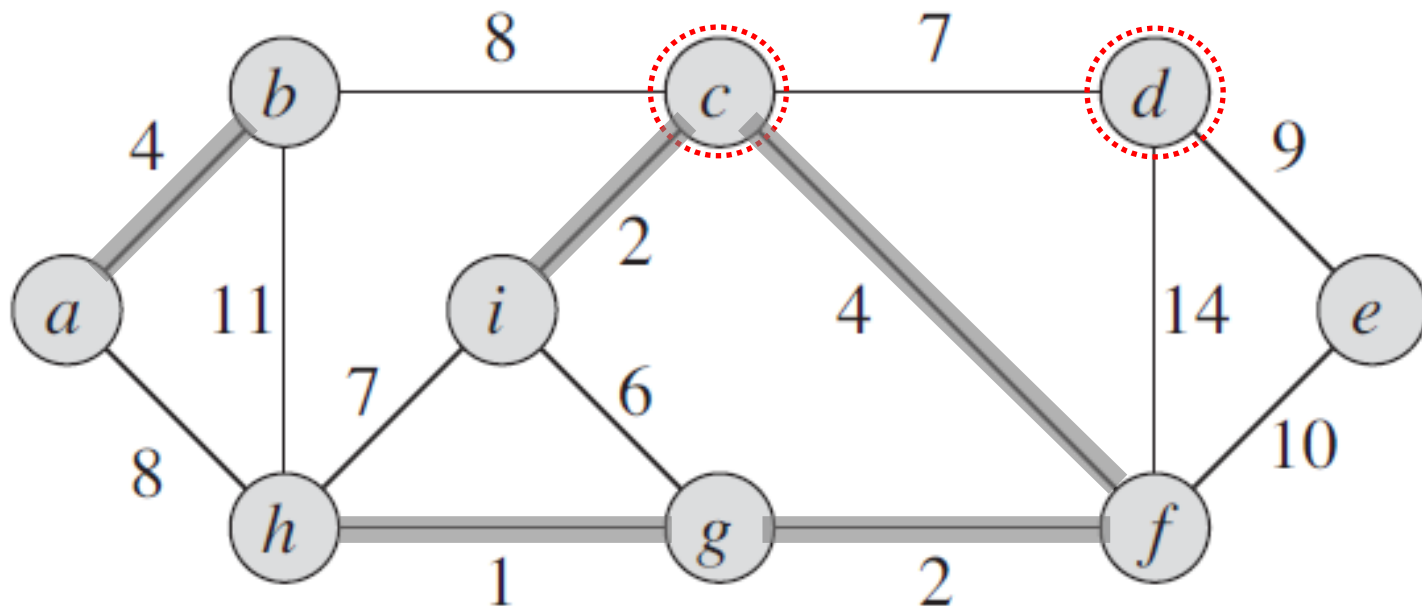
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



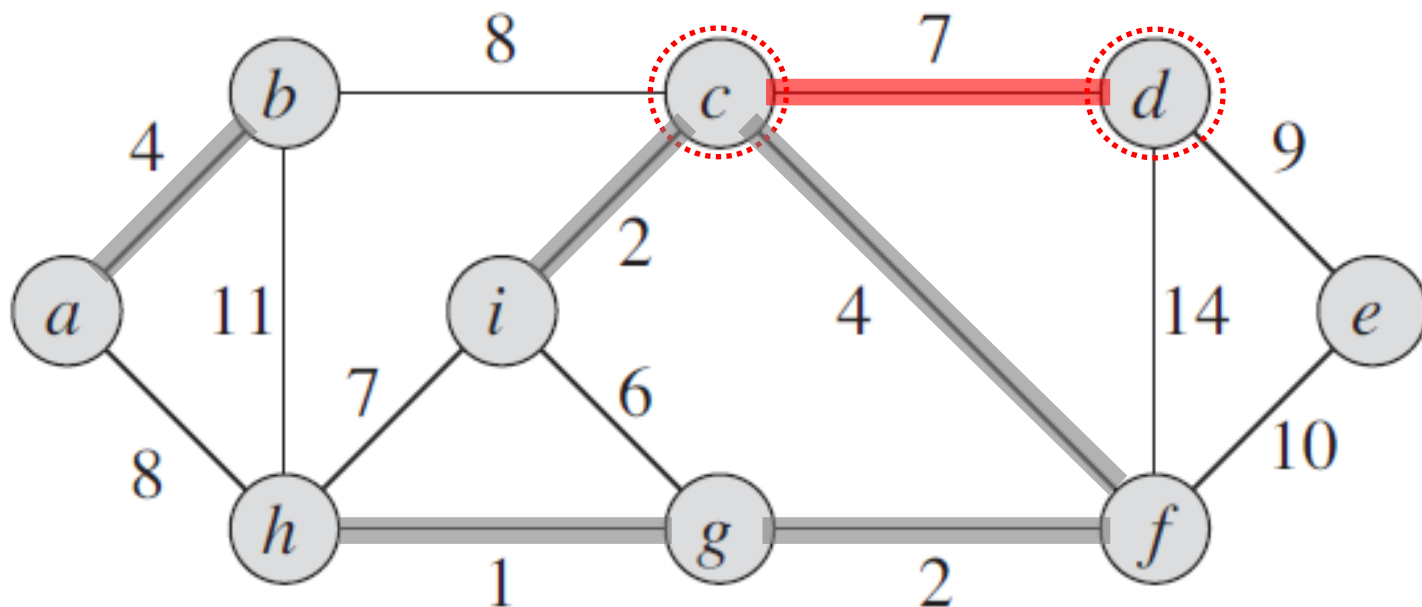
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



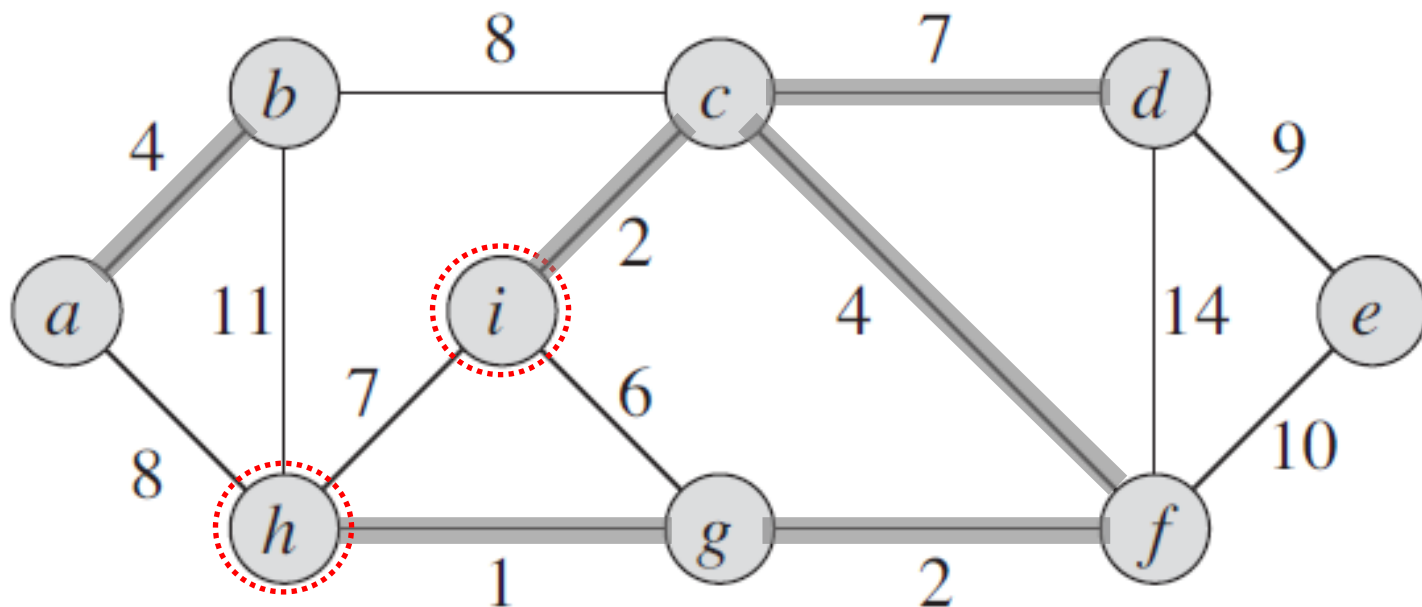
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|----|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



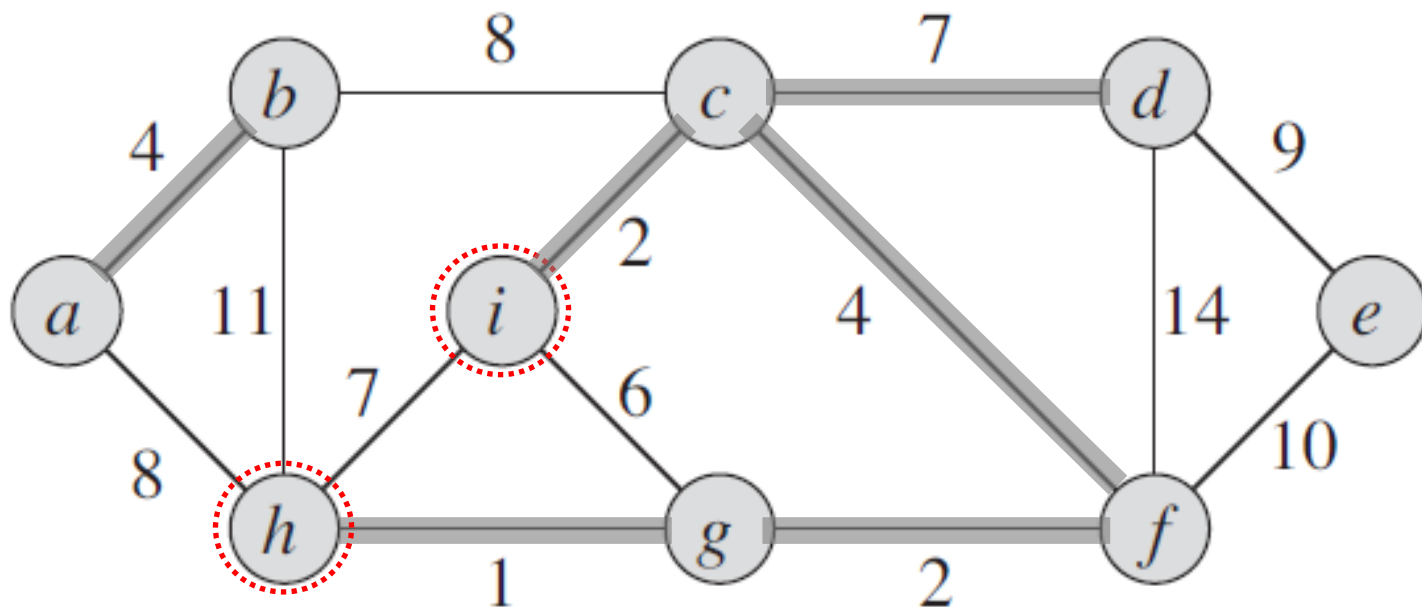
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



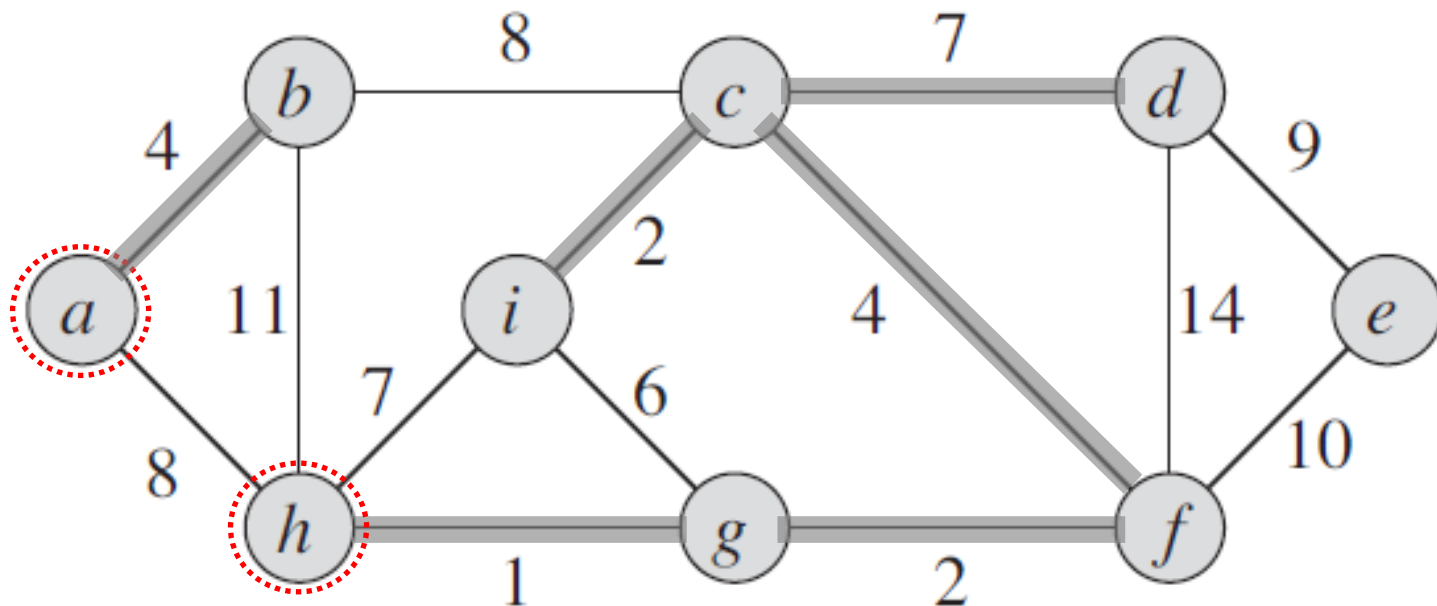
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



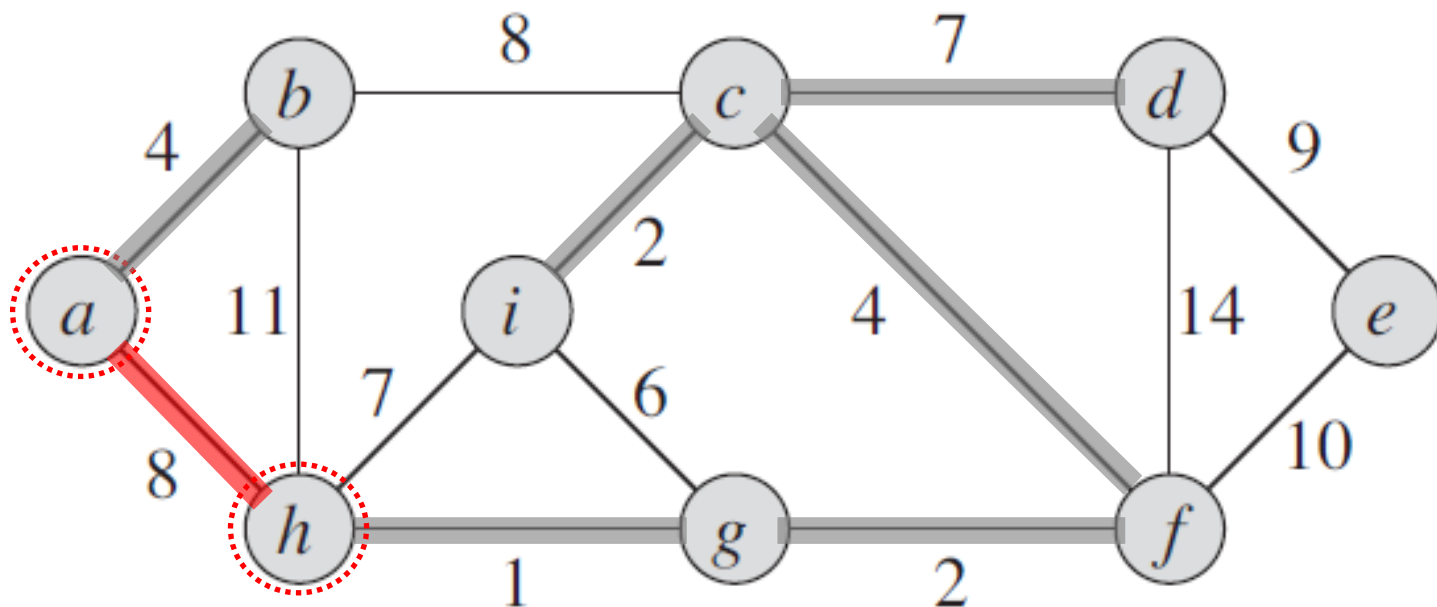
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|---------------|---------------|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



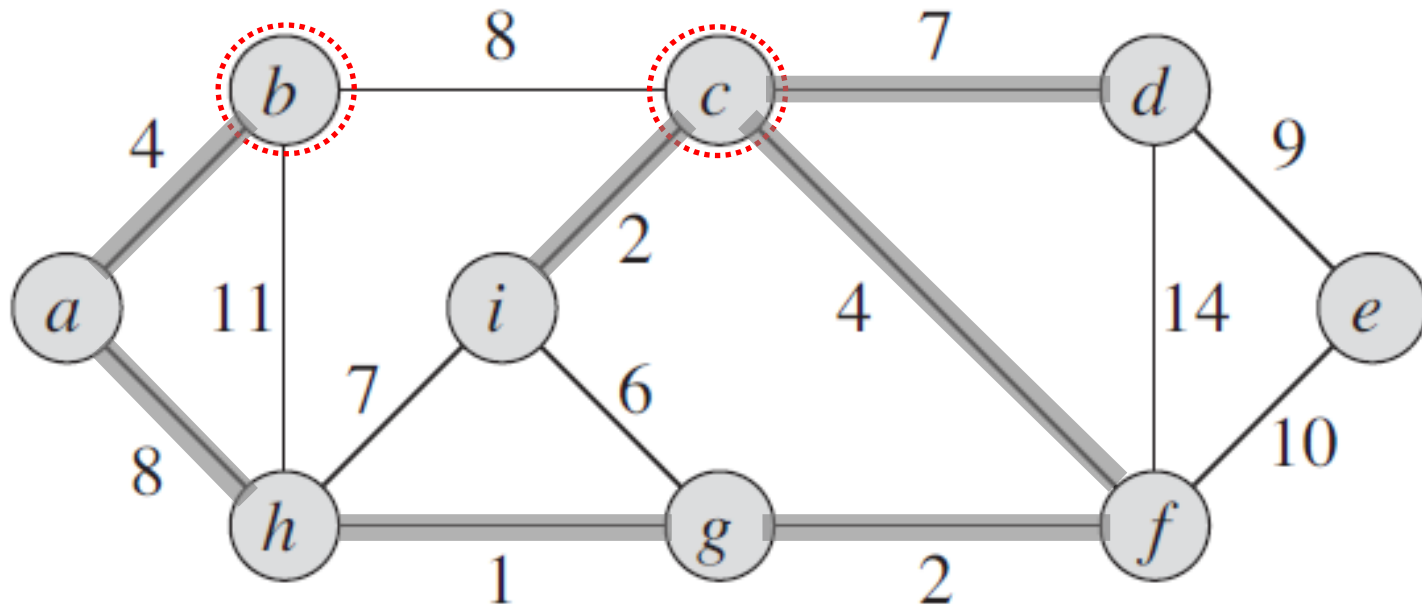

```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|----|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



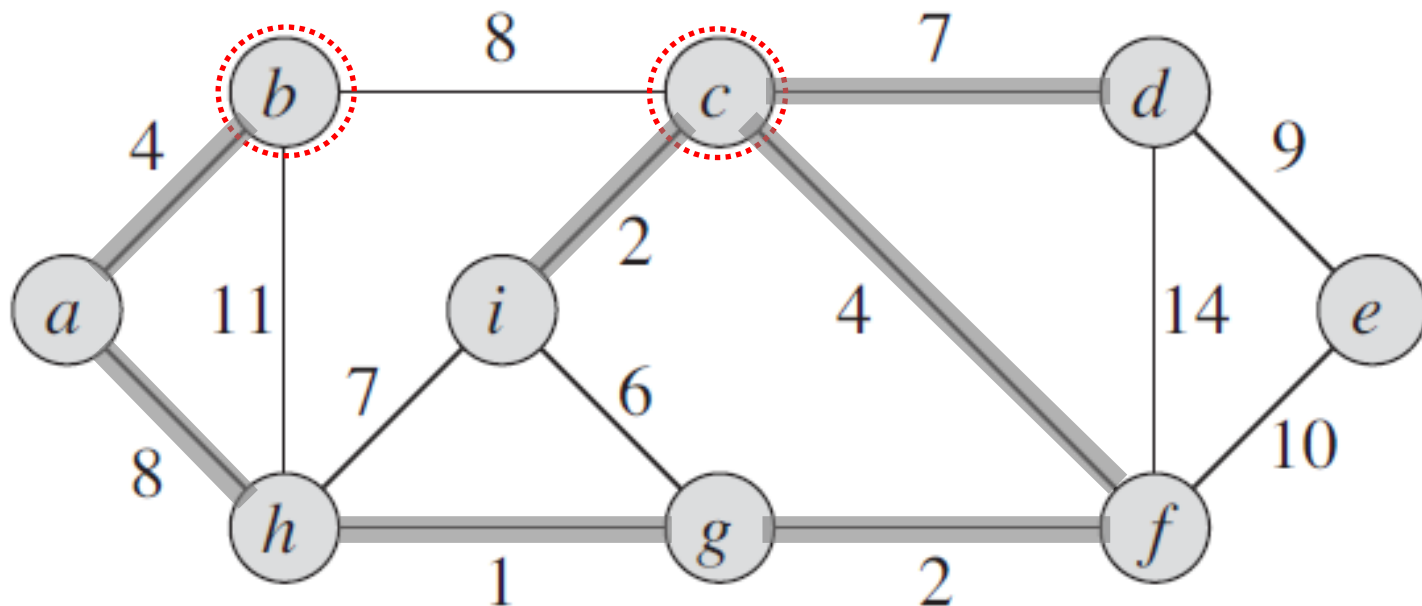
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



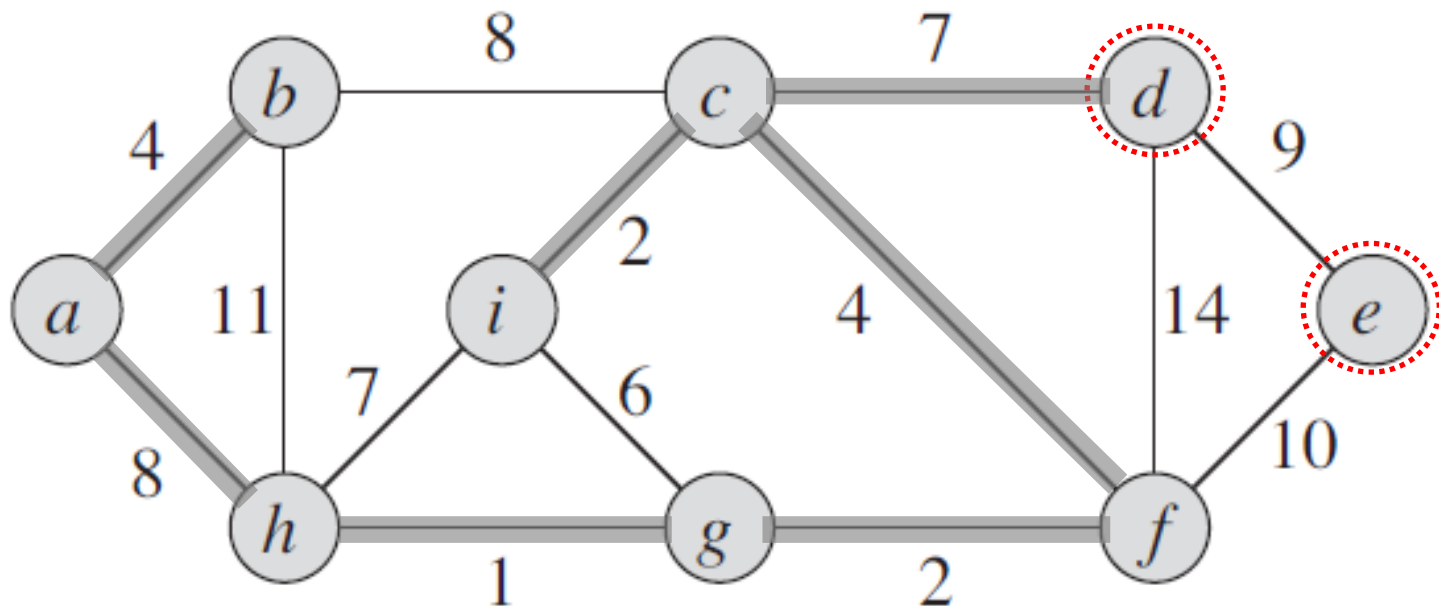
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



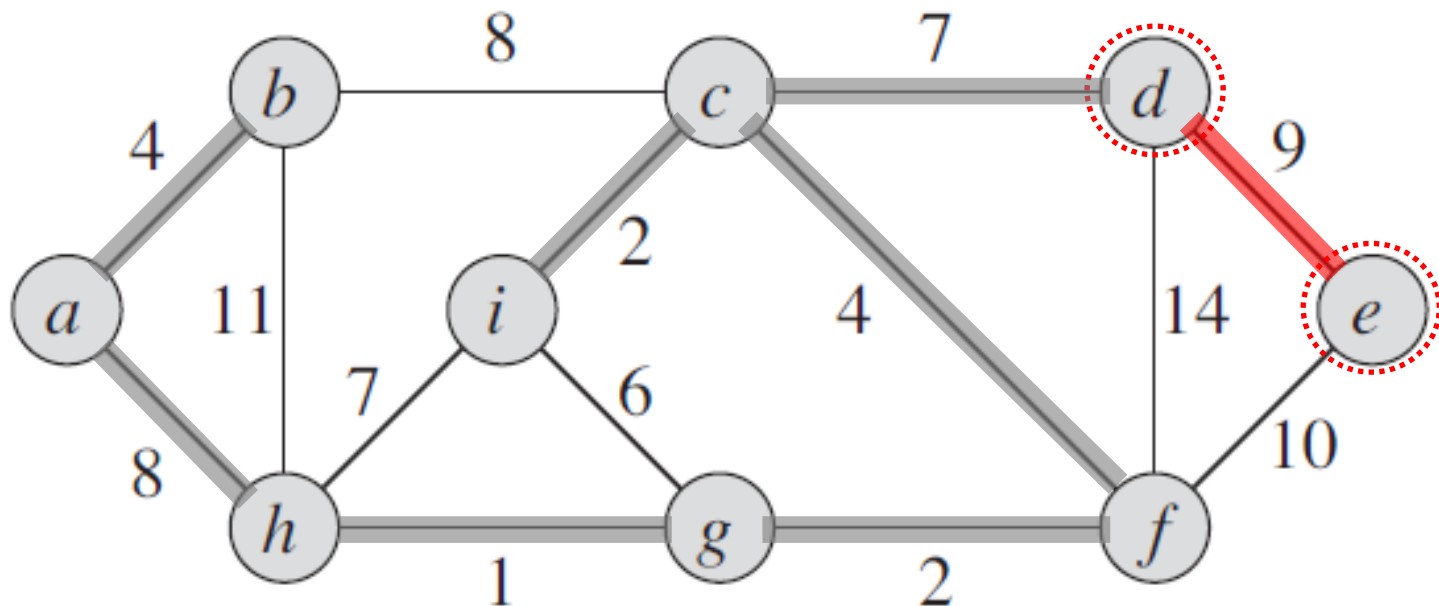
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|----|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



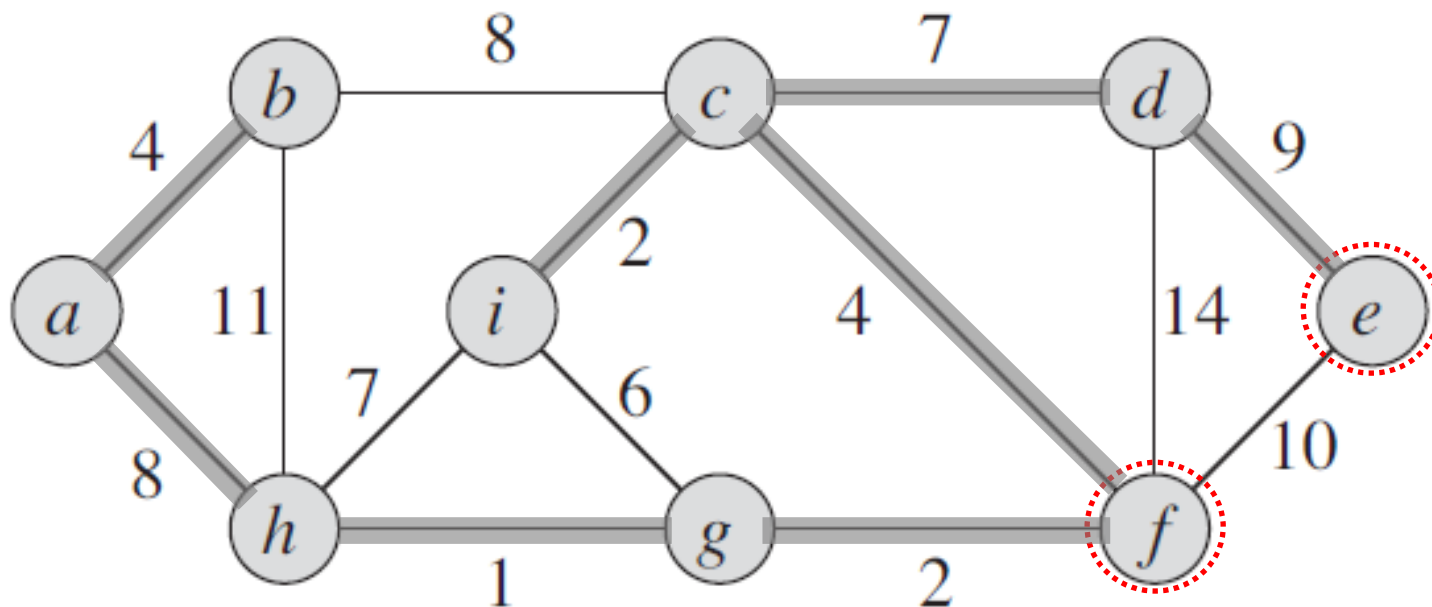
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|-----------|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



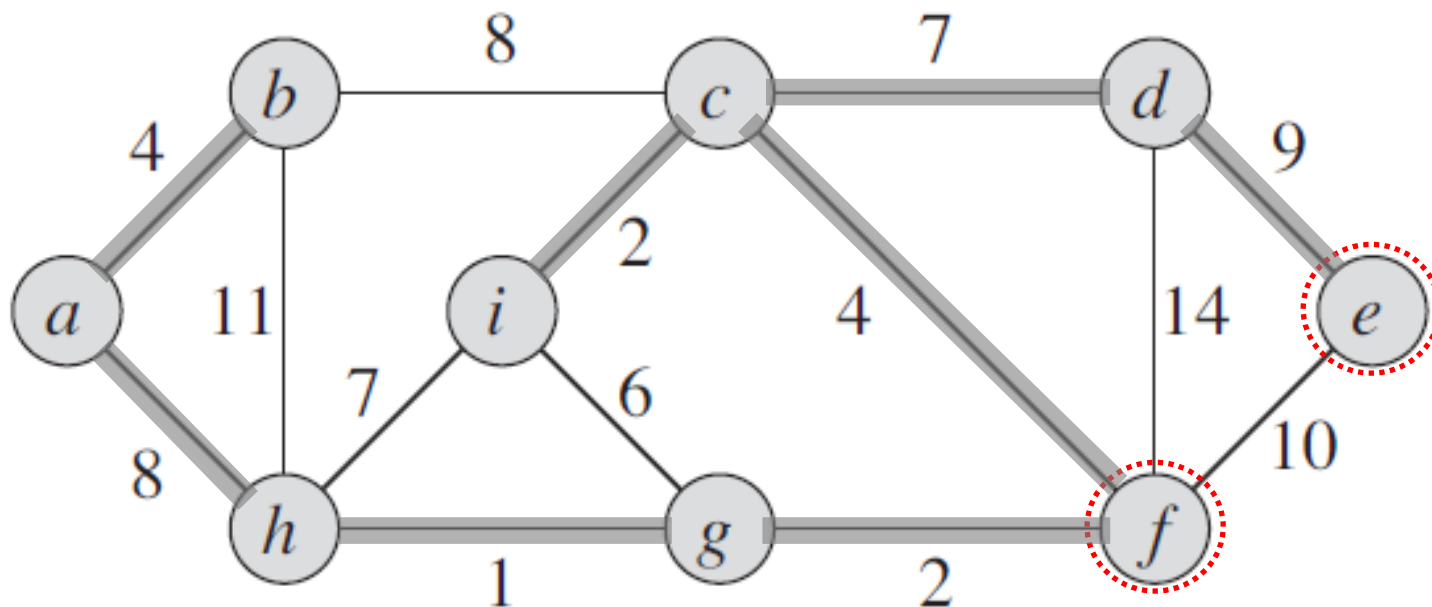
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|----|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



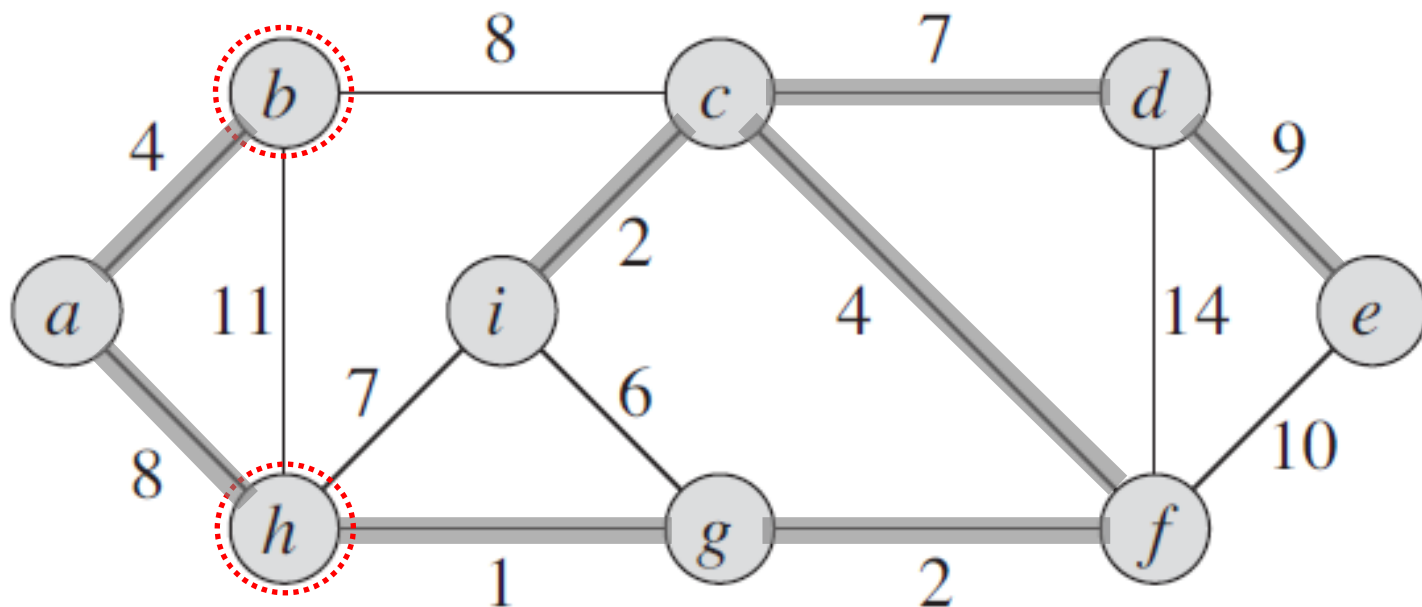
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|-----------|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



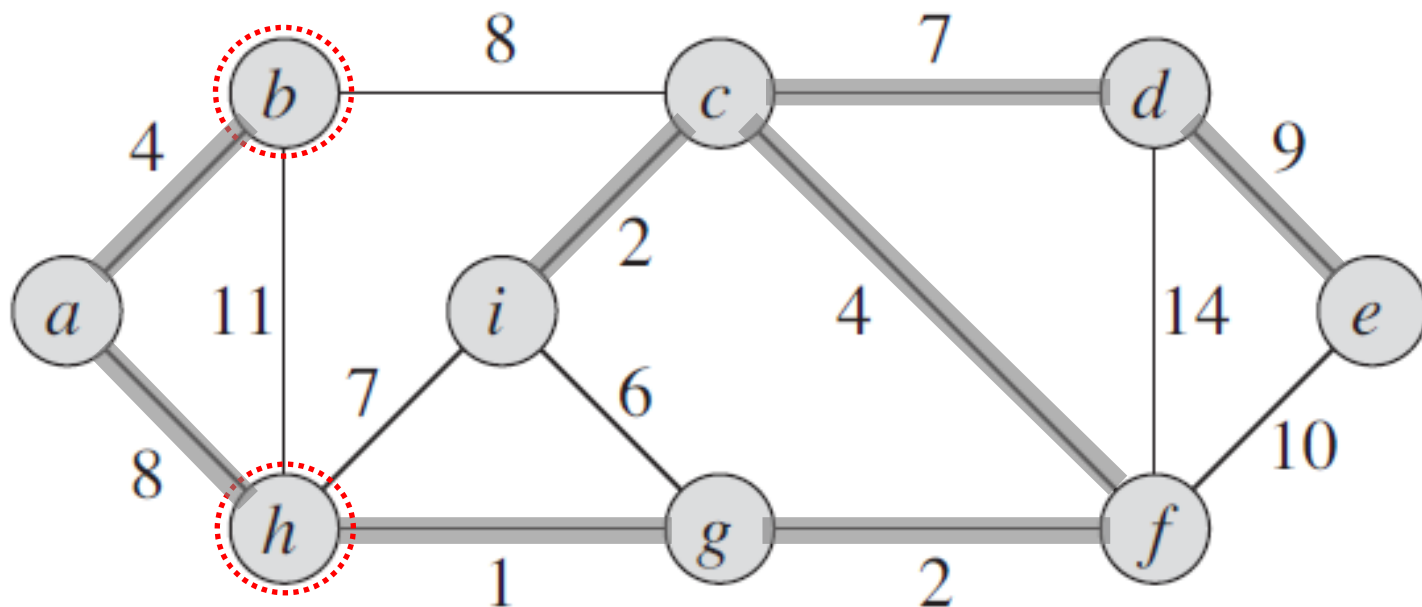
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|---------------|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



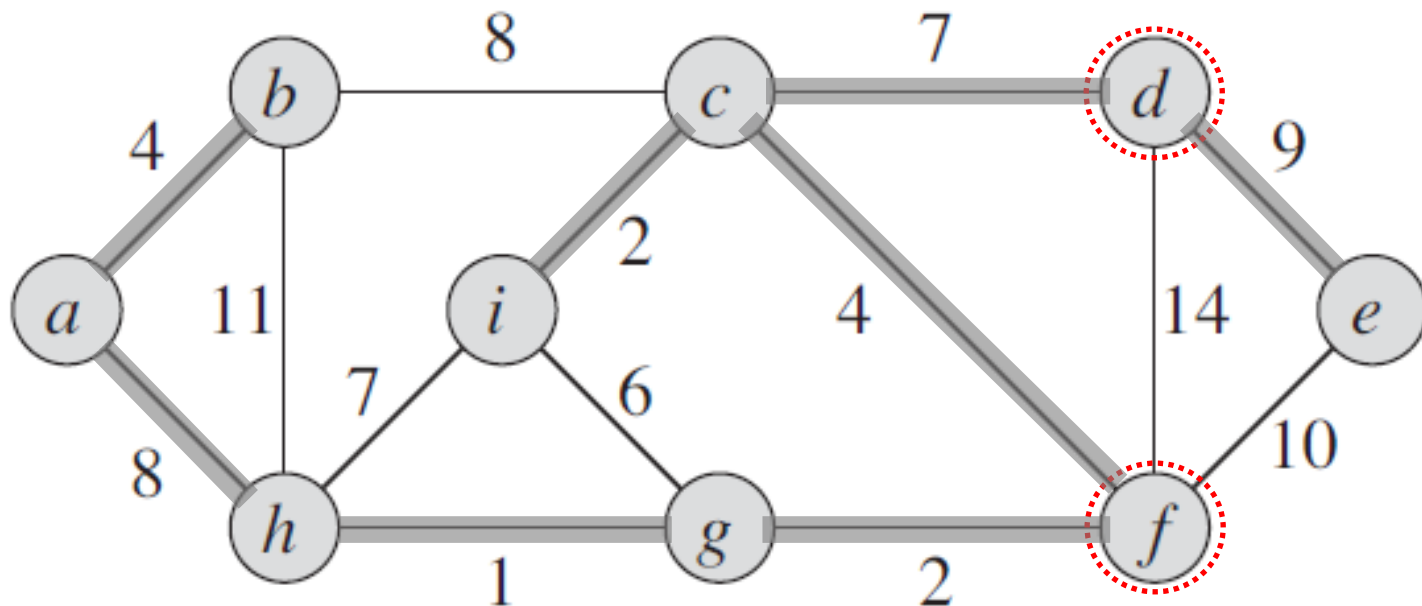

```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|---------------|----|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



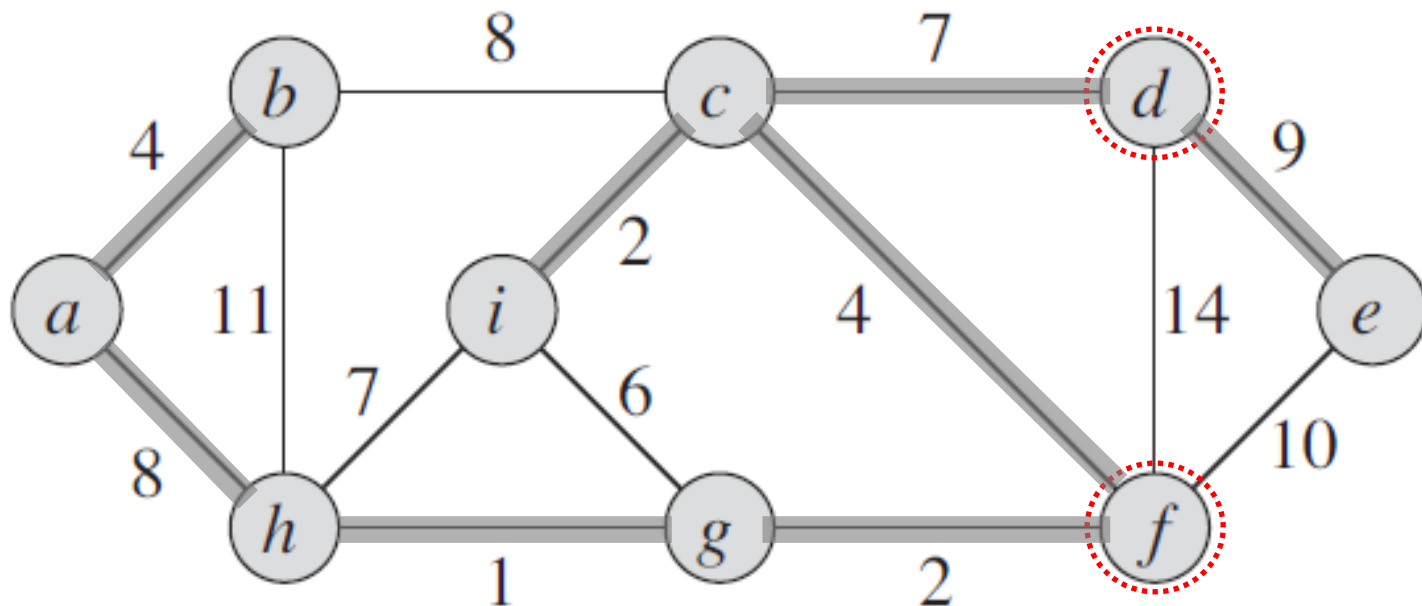
```

5 para cada aresta uv em G.E (ordenada) faça
6     se FindSet(u) != FindSet(v)
7         entao Union(u,v)
8         T = T U {uv}
9 devolva T

```

Arestas ordenadas

| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|---------------|---------------|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |

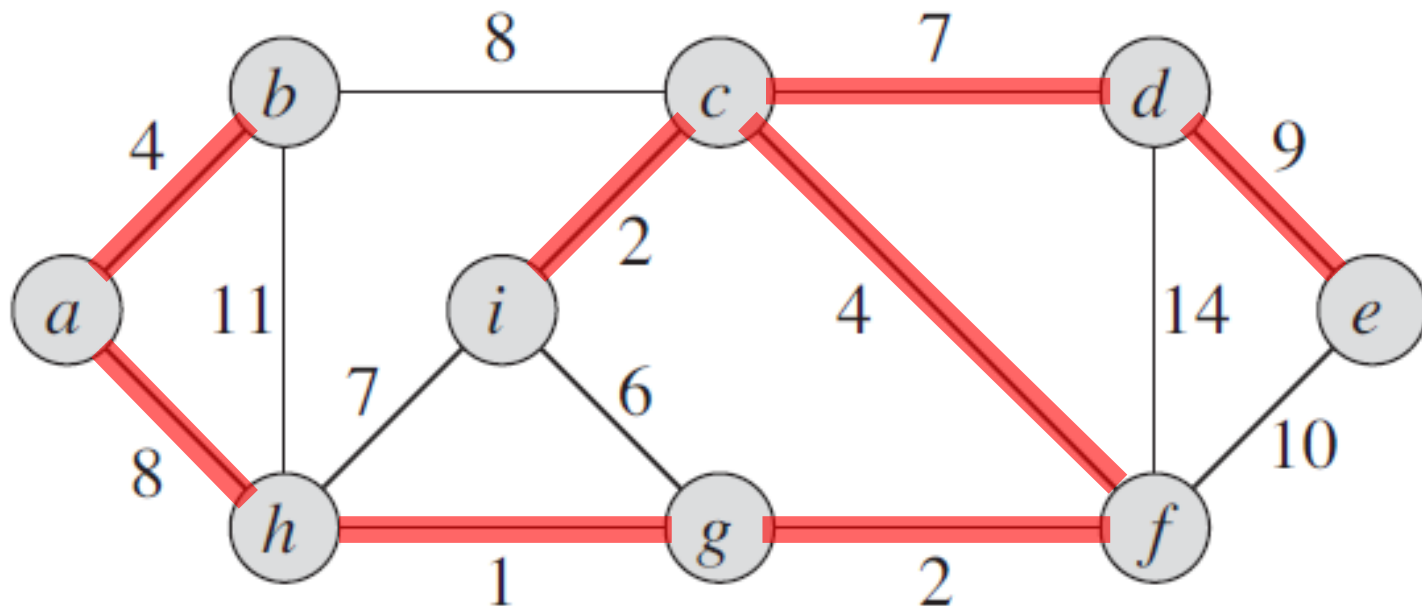


(Peso total)

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

Arestas ordenadas

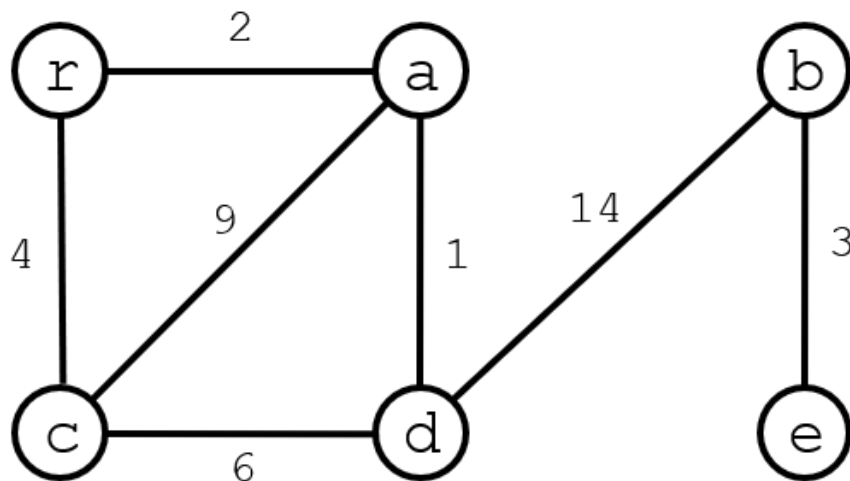
| | | | | | | | | | | | | | |
|----|----|----|----|----|---------------|----|---------------|----|---------------|----|---------------|---------------|---------------|
| gh | ci | fg | ab | cf | ig | cd | hi | ah | bc | de | ef | bh | df |
| 1 | 2 | 2 | 4 | 4 | 6 | 7 | 7 | 8 | 8 | 9 | 10 | 11 | 14 |



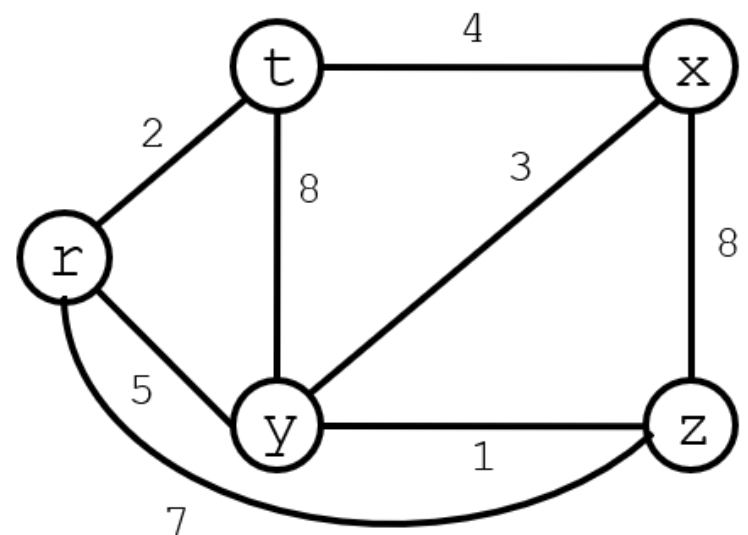
Exercícios

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

(a)



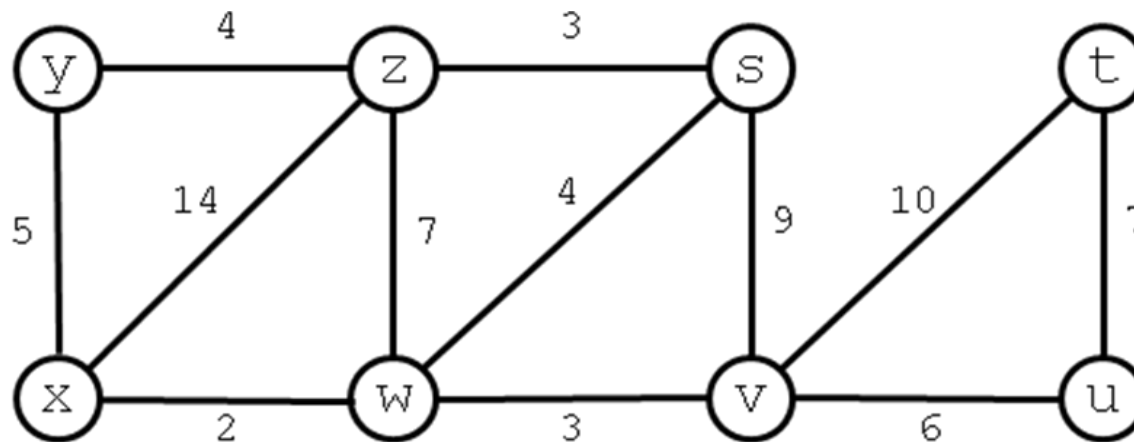
(b)



Exercícios

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

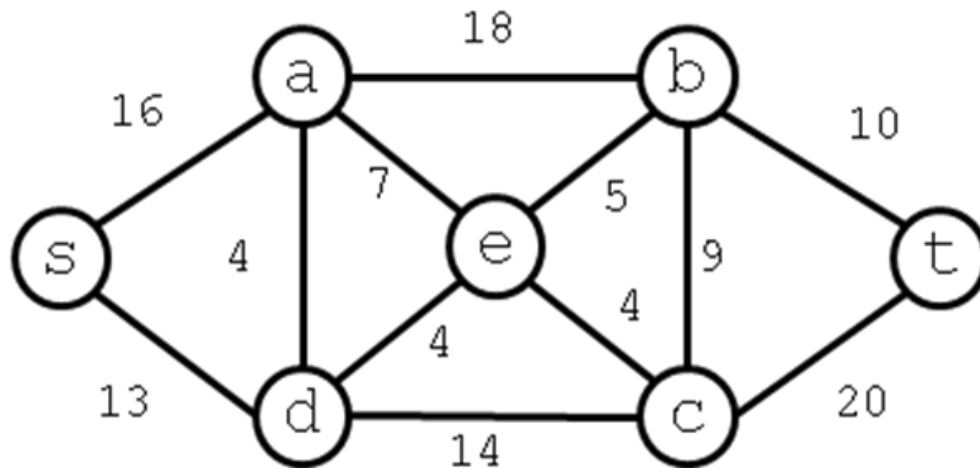
(c)



Exercícios

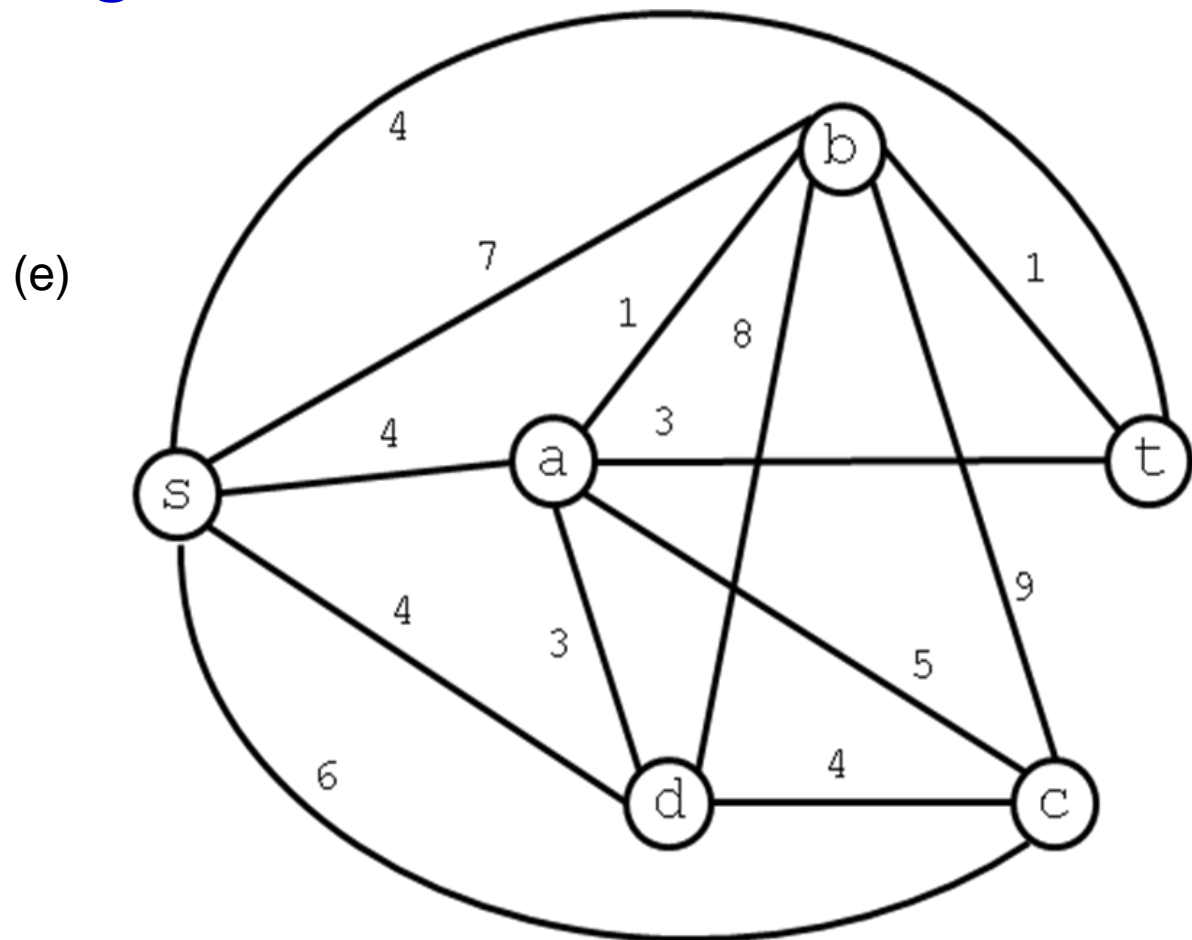
- Simule o algoritmo de **Kruskal** para calcular uma **árvore geradora mínima**:

(d)



Exercícios

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



Consumo de tempo?

MST-Kruskal (G, w)

```
1  $T = \emptyset$ 
2 para cada vértice  $u$  em  $G.V$  faça
3     MakeSet( $u$ )
4 ordenar arestas  $G.E$  por peso (crescente)
5 para cada aresta  $uv$  em  $G.E$  (ordenada) faça
6     se FindSet( $u$ )  $\neq$  FindSet( $v$ )
7         entao Union( $u, v$ )
8          $T = T \cup \{uv\}$ 
9 devolva  $T$ 
```

Consumo
de tempo

???

???

???

???

???

???

???

???

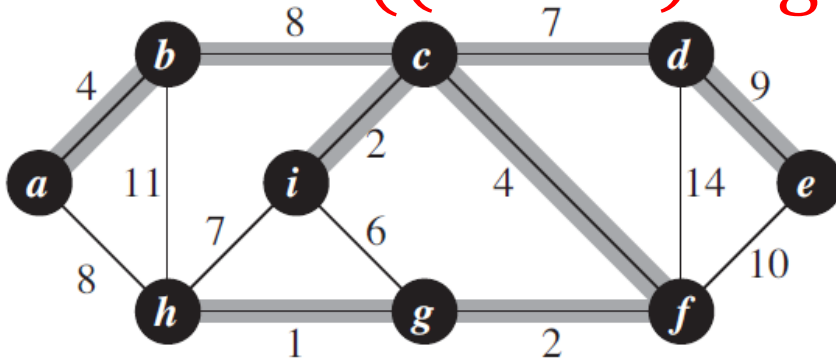
???

Total:

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

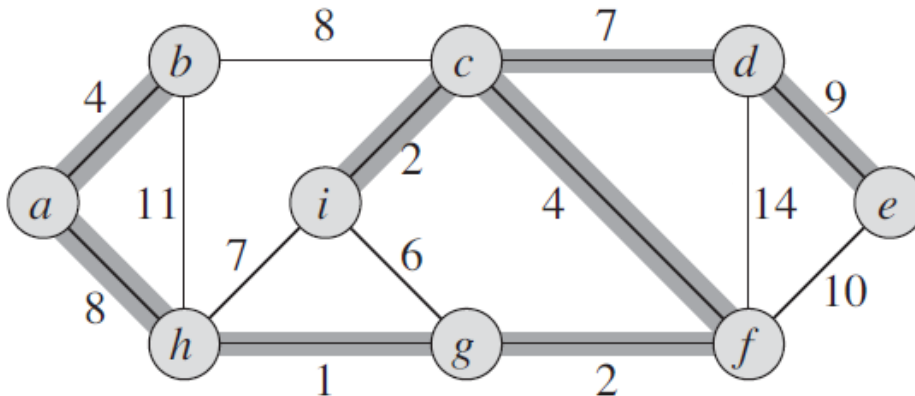
Prim vs Kruskal

- Prim $O((n + m) \log n)$



$$w(T) = 4 + 8 + 7 + 9 + 4 + 2 + 1 + 2 = 37$$

- Kruskal $\Omega(m \log m)$



$$w(T) = 1 + 2 + 2 + 4 + 4 + 7 + 8 + 9 = 37$$

Exercício Programa

- 14-mst-kruskal.py