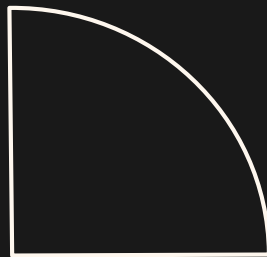
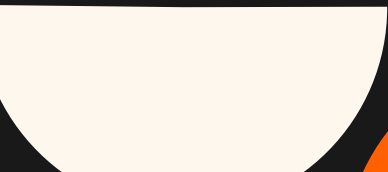




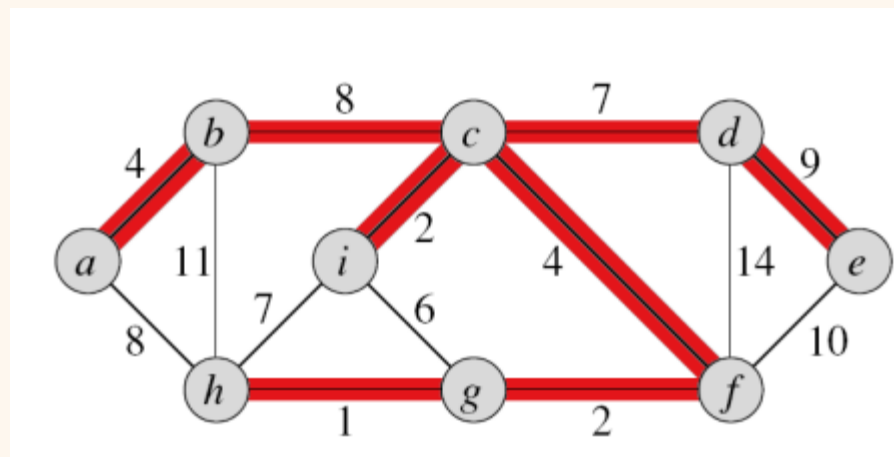
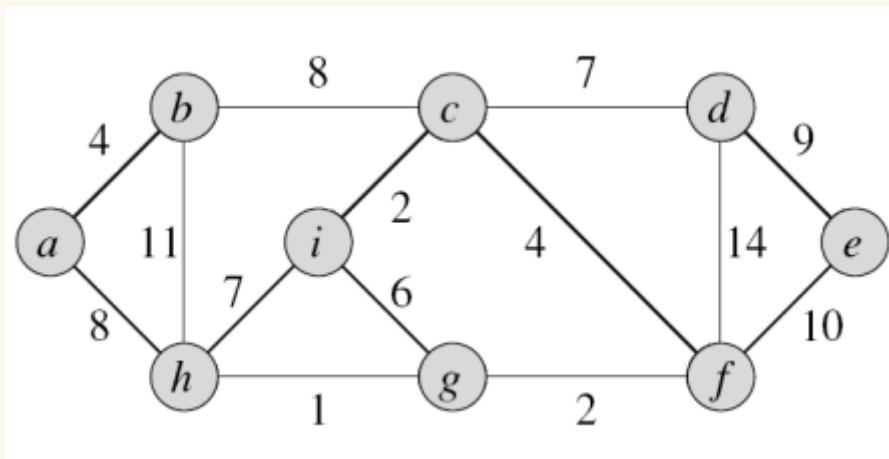
TEORIA DOS GRAFOS

Prof^a Laura Pacifico

2025 | SETEMBRO



Relembrando a aula passada...



Algoritmo de Kruskal

- Proposto por Joseph B. Kruskal em 1956.
- Formação da árvore através de inclusões de arestas, não de vértices, como em Prim.

Algoritmo de Kruskal

A

Kruskal

Ler $G=(N,M)$ $D=[d_{ij}]$ a matriz de pesos de G

Ordene as arestas em ordem  crescente de pesos d_{ij} no vetor

$H = [h_i], i = 1, 2, \dots, m$

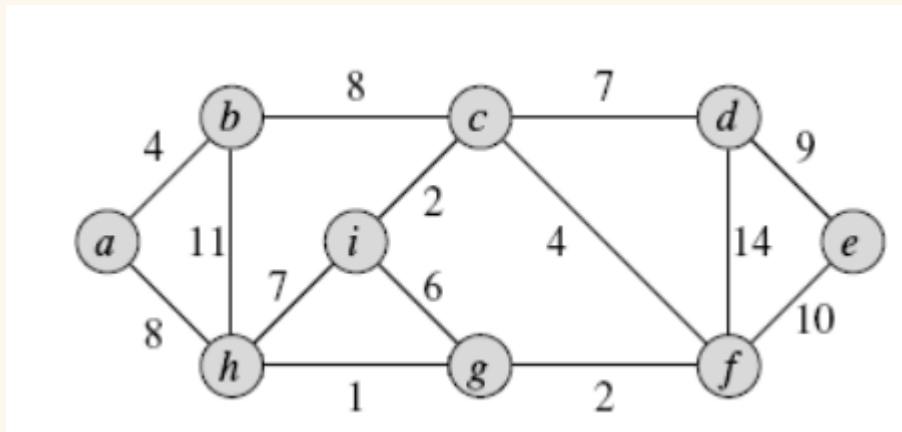
$T \leftarrow h_1$

$i \leftarrow 2$

```
Enquanto  $|T| < n$  Faça
    Se  $T \cup h_i$  é um grafo acíclico então
         $T \leftarrow T \cup h_i$ 
         $i \leftarrow i + 1$ 
    Fim_Enquanto
Escrever  $T$  {arestas da árvore geradora mínima}
```

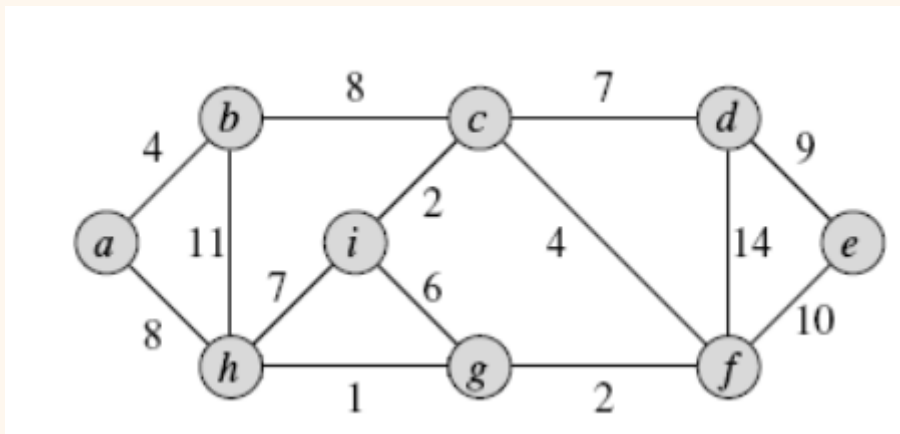
Algoritmo de Kruskal

- Considerando o grafo a seguir encontre a AGM utilizando Kruskal.



Algoritmo de Kruskal

- 1º passo: criar um conjunto/árvore para cada vértice.

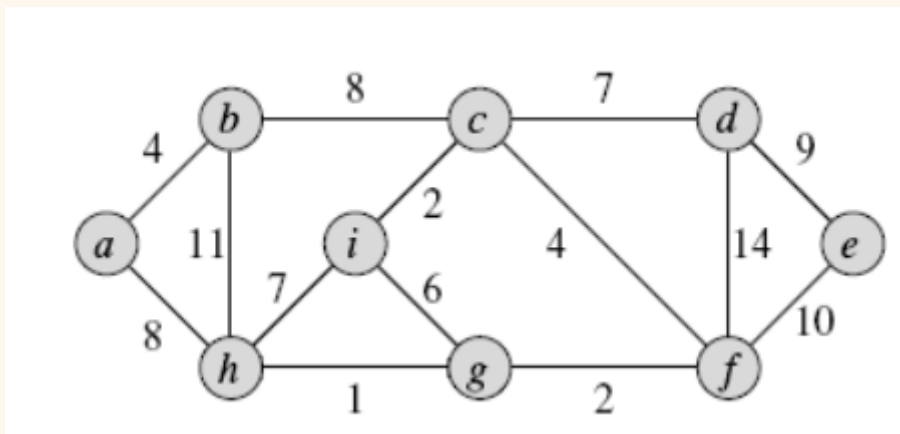


$\{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g\}, \{h\}, \{i\}\}$

Algoritmo de Kruskal

- 2º passo: ordenar as arestas do conjunto A de maneira crescente.

{{ a}, { b}, { c}, { d}, { e}, { f }, { g}, { h}, { i } }



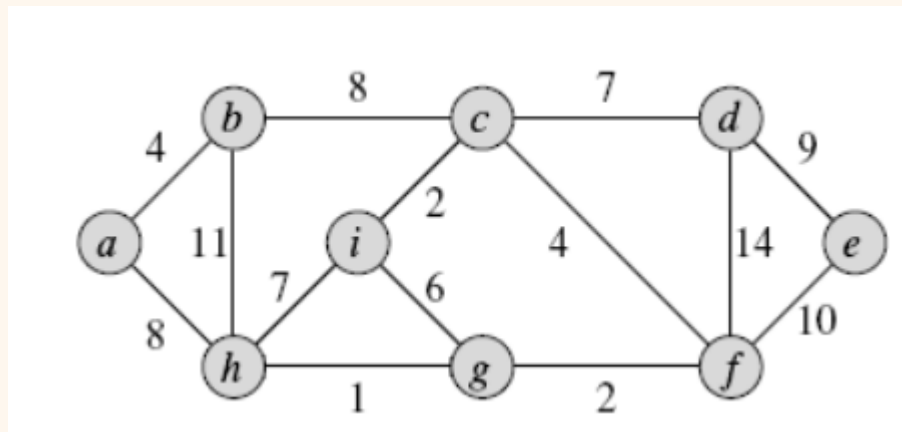
A' (g, h); (c, i); (f, g); (a, b); (c, f); (g, i); (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a}, { b}, { c}, { d}, { e}, { f }, { g}, { h}, { i } }

g e h pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); (g, i); (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

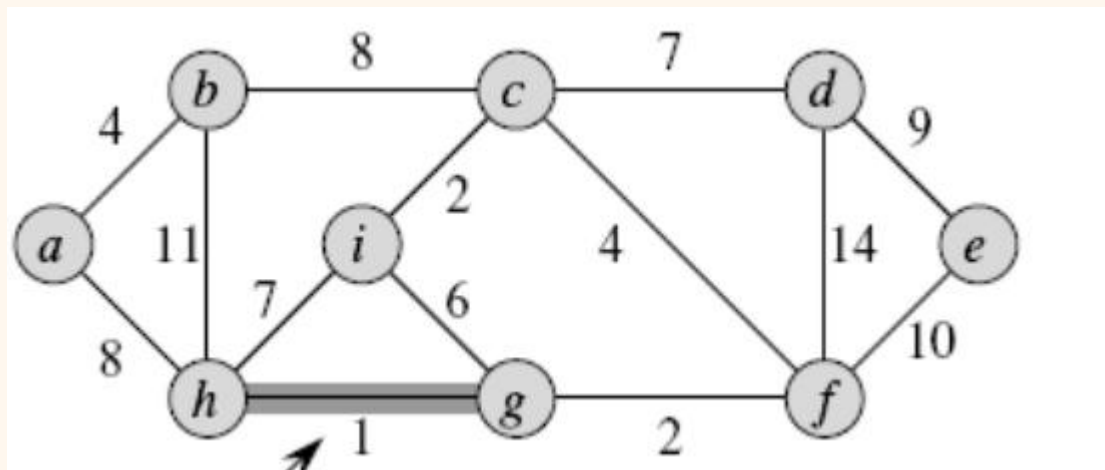
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

$\{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{f\}, \{g, h\}, \{i\}\}$

g e h pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de g e h e adição da
aresta na AGM**



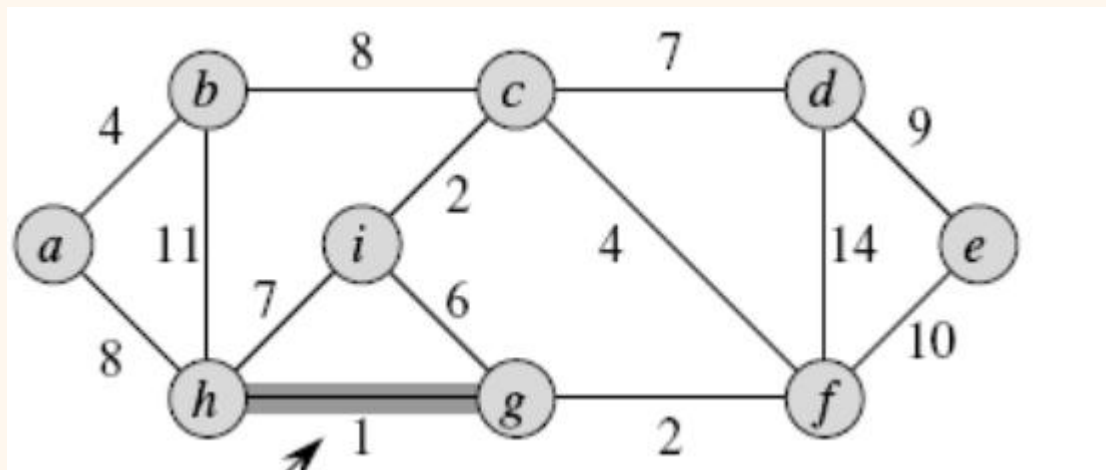
A' (g, h); (c, i); (f, g); (a, b); (c, f); (g, i); (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a }, { b }, { c }, { d }, { e }, { f }, { g, h }, { i } }

c e i pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); (g, i); (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

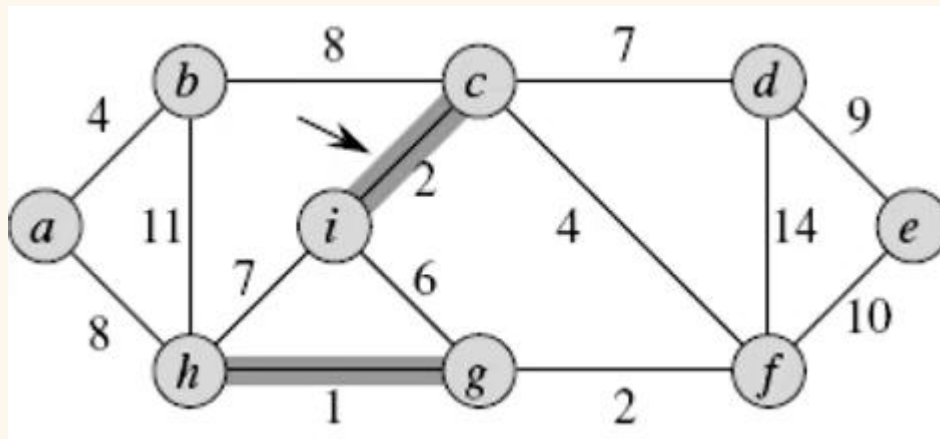
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a }, { b }, { c, i }, { d }, { e }, { f }, { g, h }}

c e i pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de c e i e adição da
aresta na AGM**



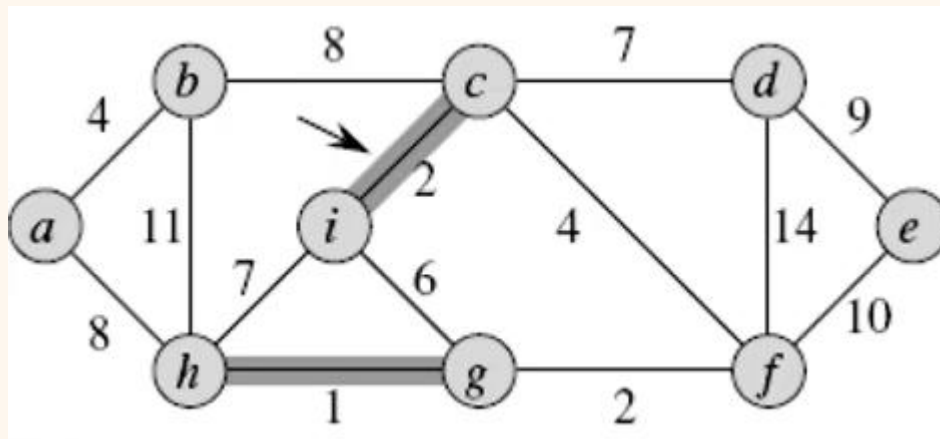
A' (g, h); (c, i); (f, g); (a, b); (c, f); (g, i); (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a }, { b }, { c, i }, { d }, { e }, { f }, { g, h }}

f e g pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

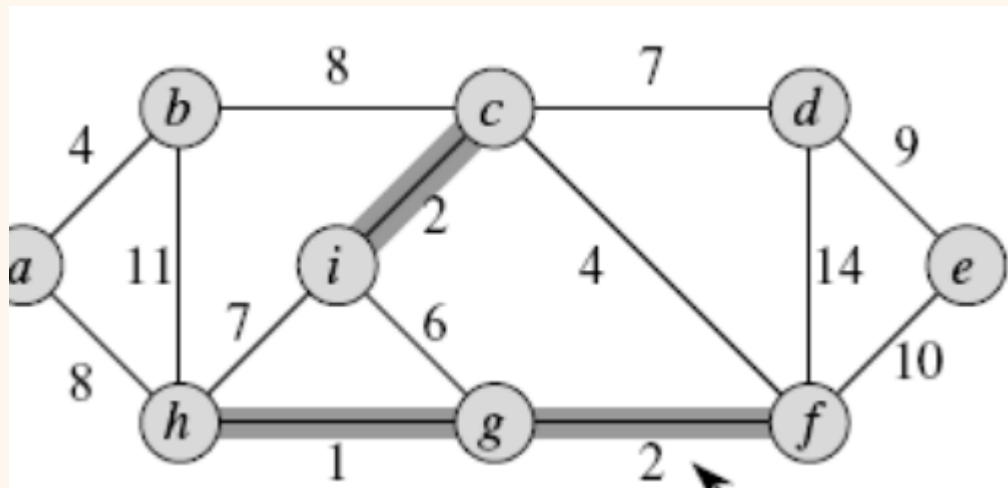
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a }, { b }, { c, i }, { d }, { e }, { f, g, h }}

f e g pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de f e g e adição da
aresta na AGM**



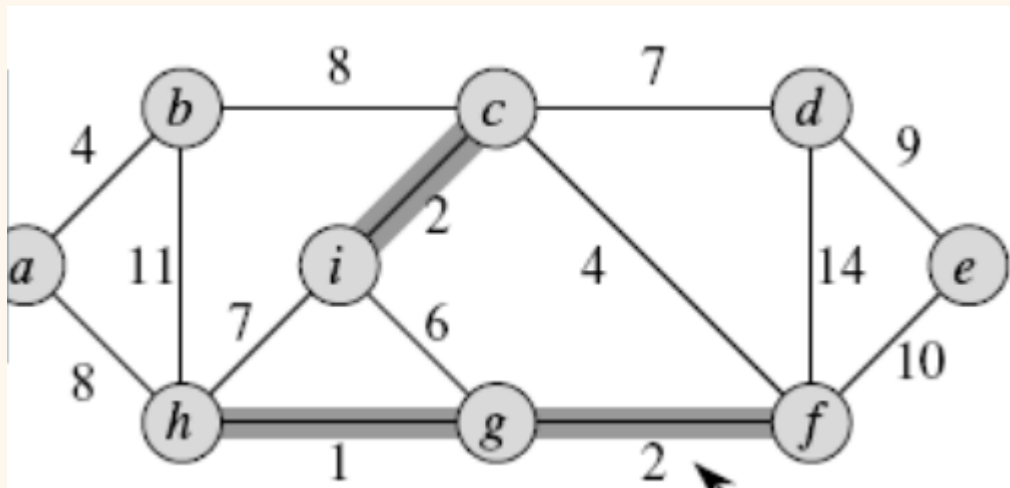
A' (g, h); (c, i); (f, g); (a, b); (c, f); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a }, { b }, { c, i }, { d }, { e }, { f, g, h }}

a e b pertencem a
mesma árvore na
floresta?



A' (g, h); (c,i);(f,g);(a,b);(c, f);(g i); (c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

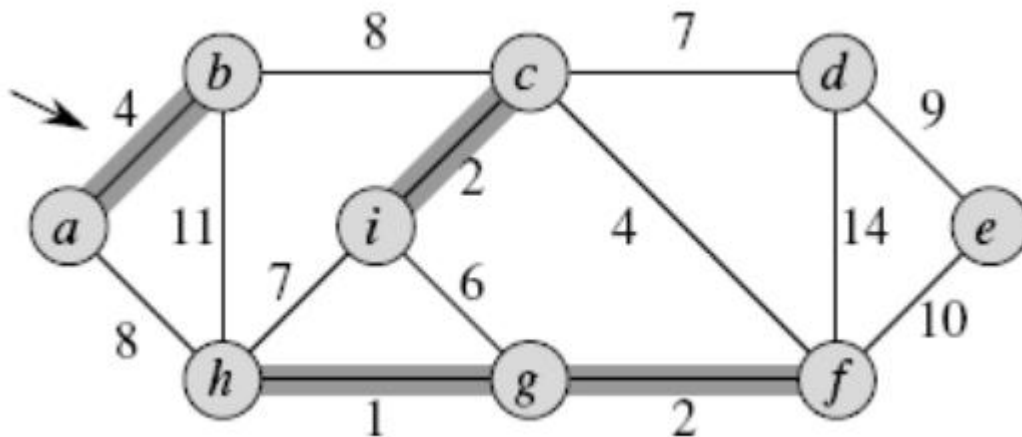
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { c, i }, { d}, { e}, { f, g, h}}

a e b pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de a e b e adição da
aresta na AGM**



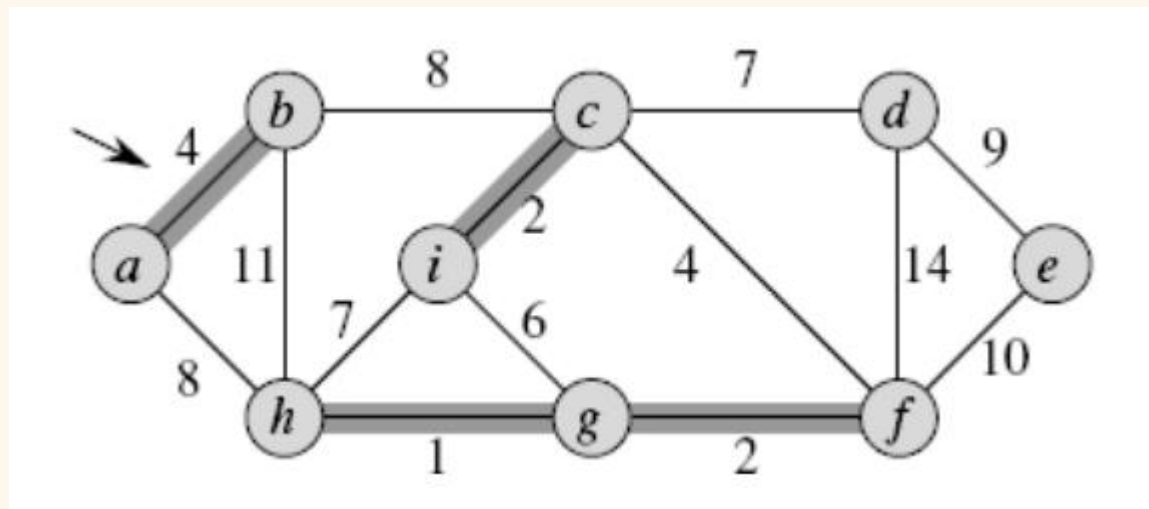
A' (g, h); (c,i); (f,g);(a,b);(c, f);(g i); (c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { c, i }, { d}, { e}, { f, g, h}}

c e f pertencem a
mesma árvore na
floresta?



A' (g, h); (c,i);(f,g);(a,b);(c, f);(g i); (c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

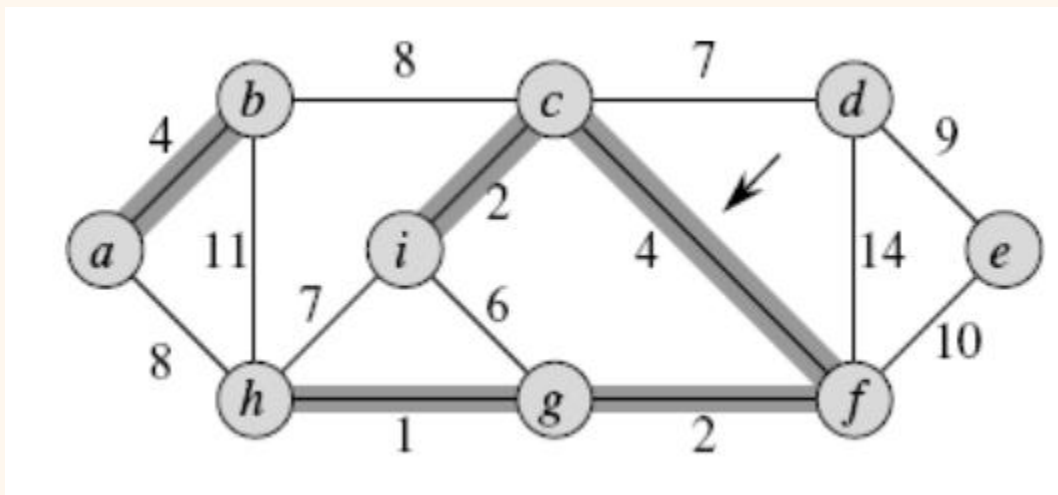
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { d}, { e}, {c, f, g, h,i}}

c e f pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de c e f e adição da
aresta na AGM**



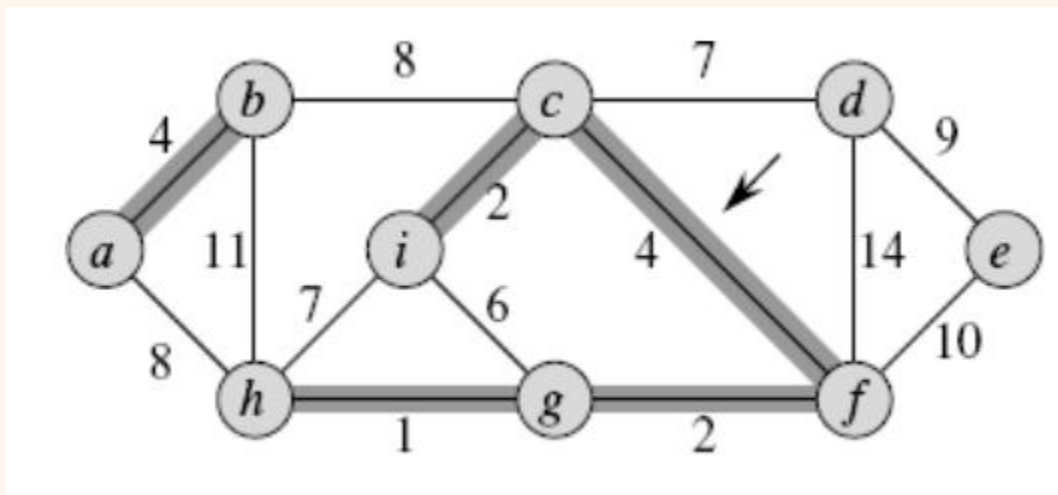
A' (g, h); (c,i);(f,g);(a,b);(c, f);(g i); (c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { d}, { e}, {c, f, g, h,i}}

g e i pertencem a
mesma árvore na
floresta?



A' (g, h); (c,i); (f,g);(a,b);(c, f);(g i); (c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

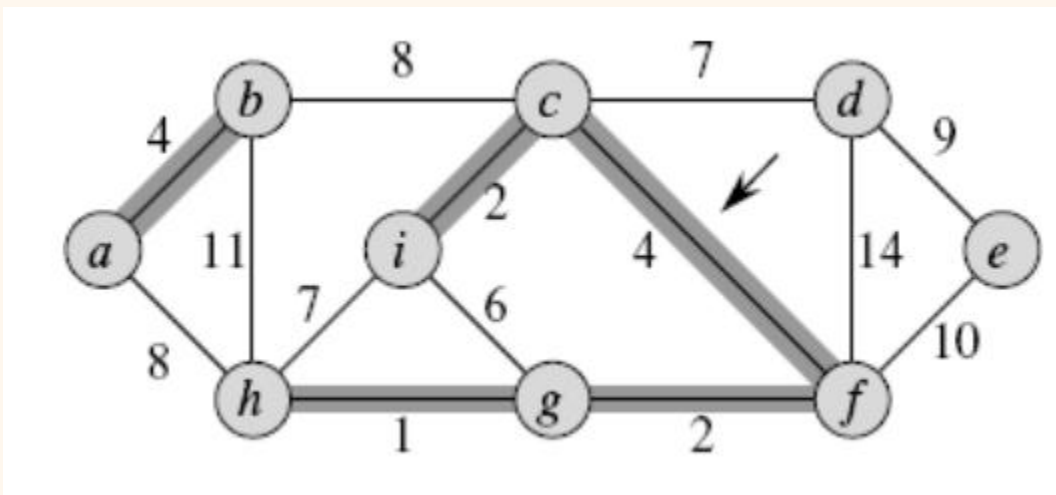
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { d}, { e}, {c, f, g, h, i}}

g e i pertencem a
mesma árvore na
floresta?

(g,i) fecha um ciclo. Isso é identificado porque g e i pertencem a mesma árvore na estrutura auxiliar ‘floresta’.



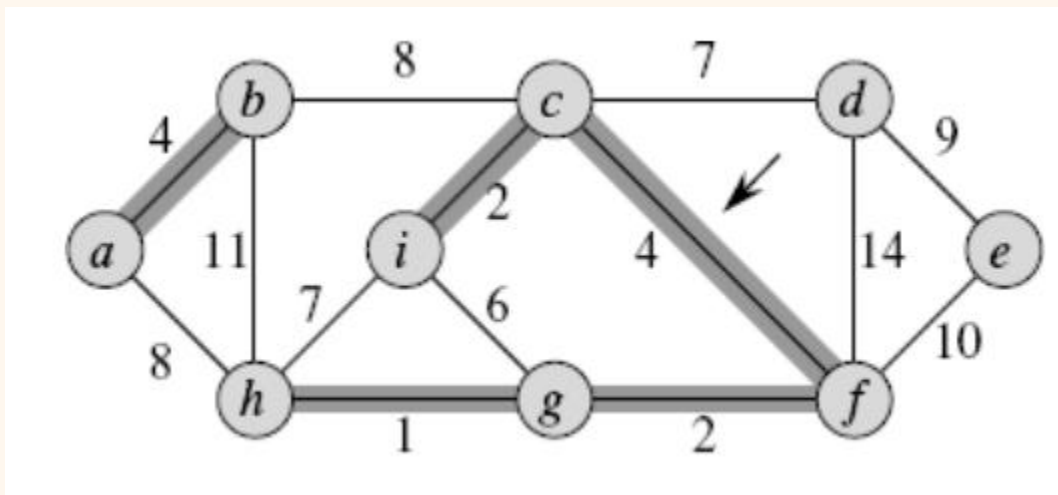
A' (g, h); (c,i);(f,g);(a,b);(c, f);(~~g,i~~);(c, d);(h,i);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { d}, { e}, {c, f, g, h, i}}

c e d pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~ (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

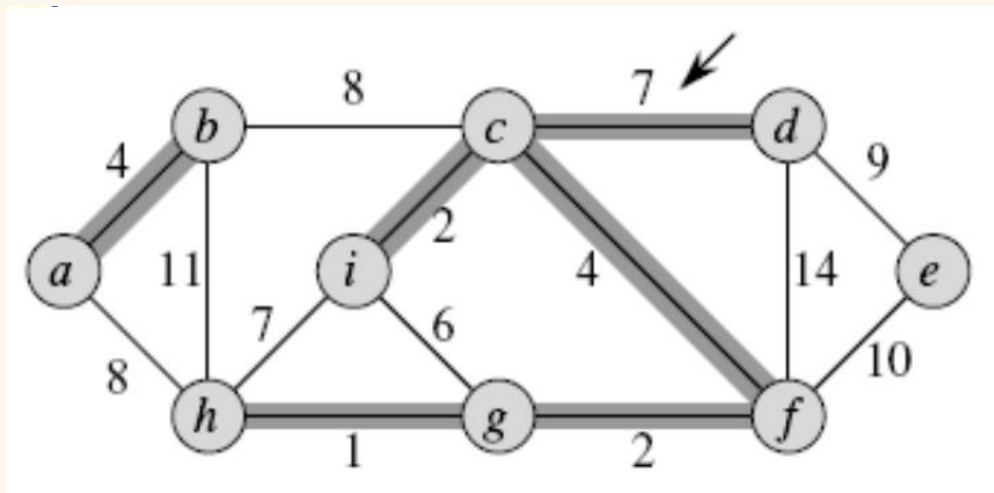
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { e}, {c, d, f, g, h, i}}

c e d pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de c e d e adição da
aresta na AGM**



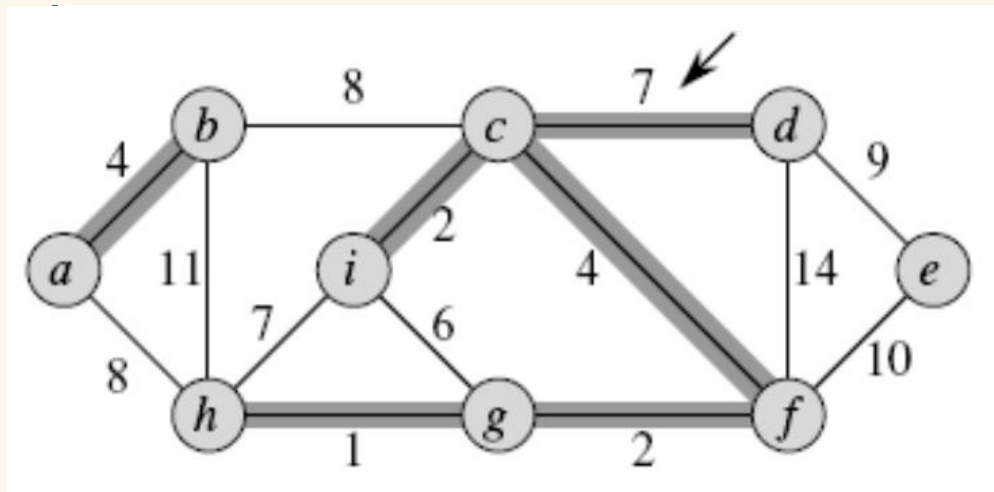
A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~ (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { e}, {c, d, f, g, h, i}}

h e i pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~ (c, d); (h, i); (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

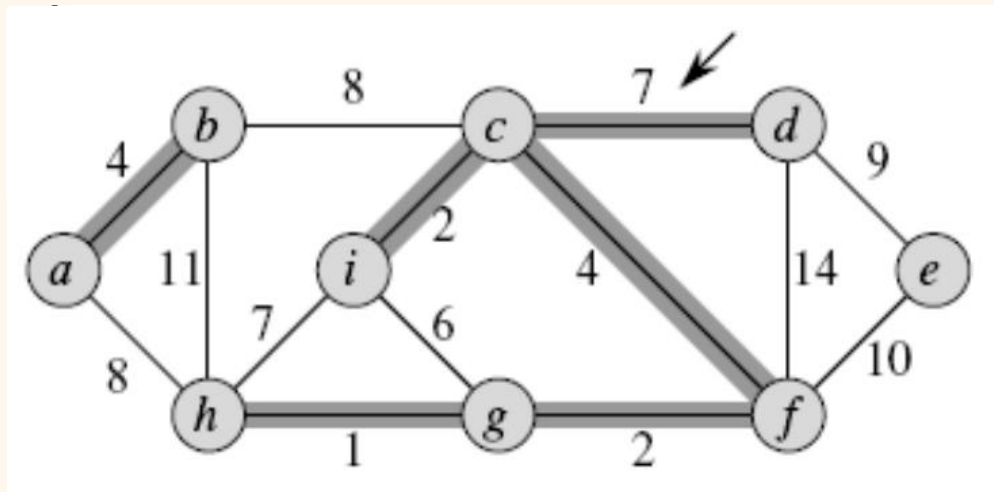
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ a , b}, { e}, {c, d, f, g, h, i}}

h e i pertencem a
mesma árvore na
floresta?

(h,i) fecha um ciclo.



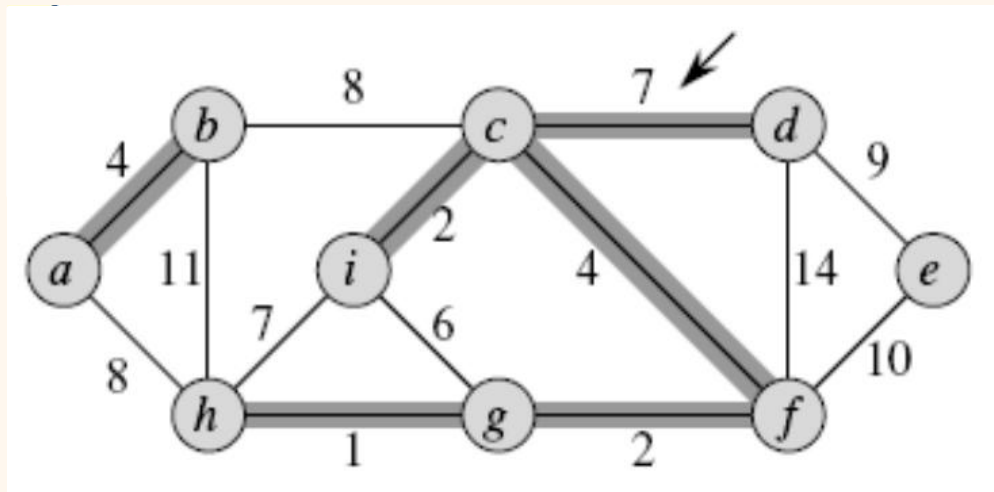
A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{a, b}, { e}, {c, d, f, g, h, i}}

a e h pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; ~~(c, d)~~; ~~(h, i)~~; (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

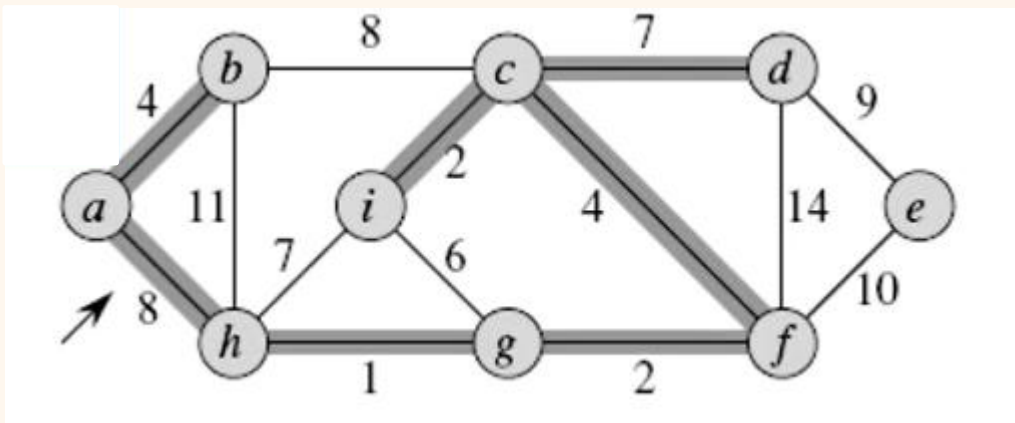
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

$$\{\{e\}, \{a, b, c, d, f, g, h, i\}\}$$

a e h pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de a e h e adição da
aresta na AGM**



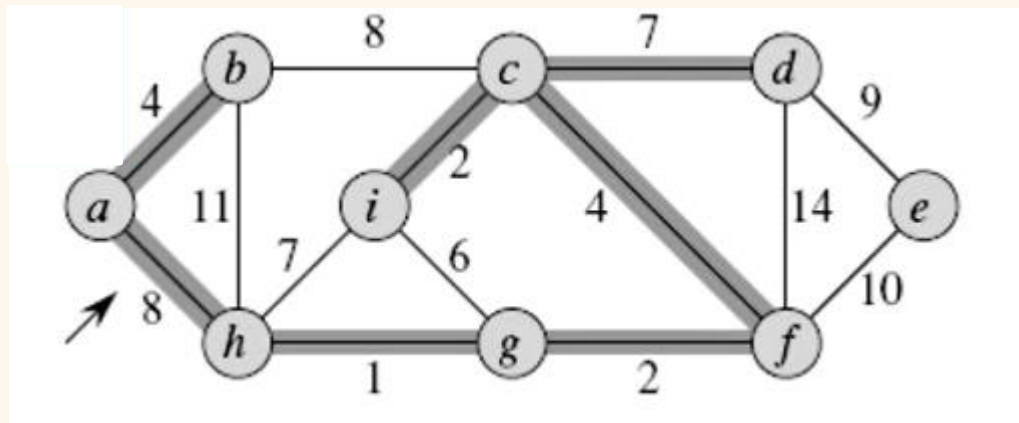
A' (g, h); (c,i); (f,g);(a,b);(c, f);(~~g,i~~); (c, d);(~~h,i~~);(a,h);(b, c);(d, e);(e, f);(b,h);(d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ e}, {a, b, c, d, f, g, h, i}}

b e c pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); (b, c); (d, e); (e, f); (b, h); (d, f)

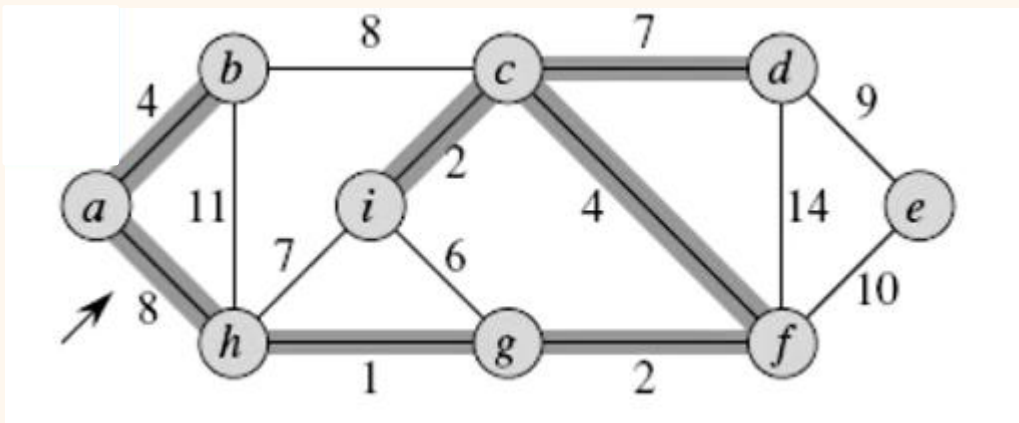
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ e}, {a, b, c, d, f, g, h, i}}

b e c pertencem a
mesma árvore na
floresta?

(b,c) fecha um ciclo.



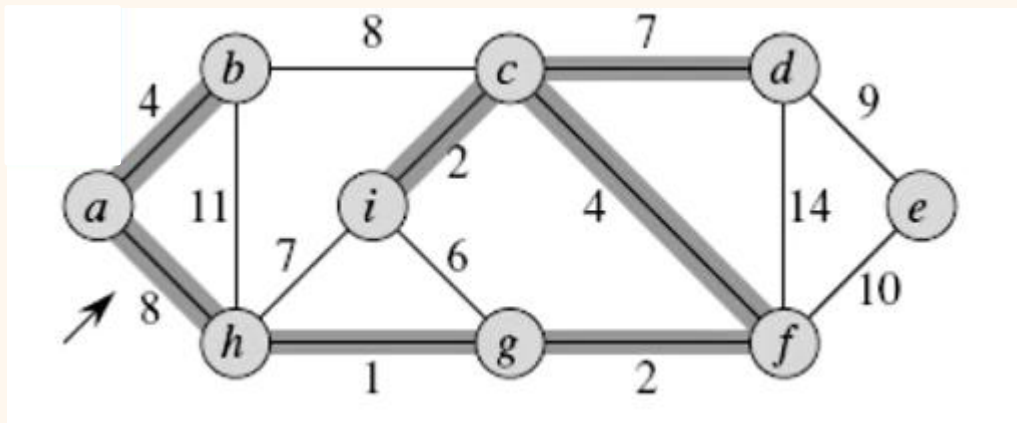
A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); (e, f); (b, h); (d, f)

Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{{ e}, {a, b, c, d, f, g, h, i}}

d e e pertencem a
mesma árvore na
floresta?



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); (e, f); (b, h); (d, f)

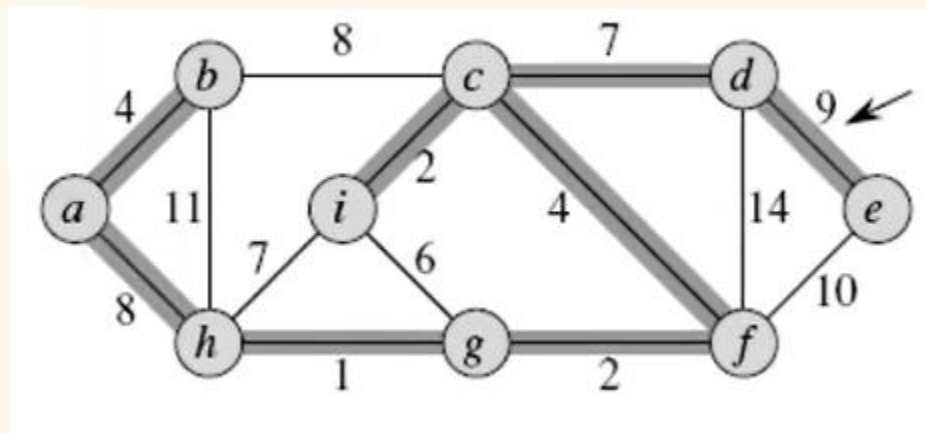
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{ {a, b, c, d, e, f, g, h, i}

d e e pertencem a
mesma árvore na
floresta?

**Não! Então:
União das árvores
de d e e e adição da
aresta na AGM**



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); (e, f); (b, h); (d, f)

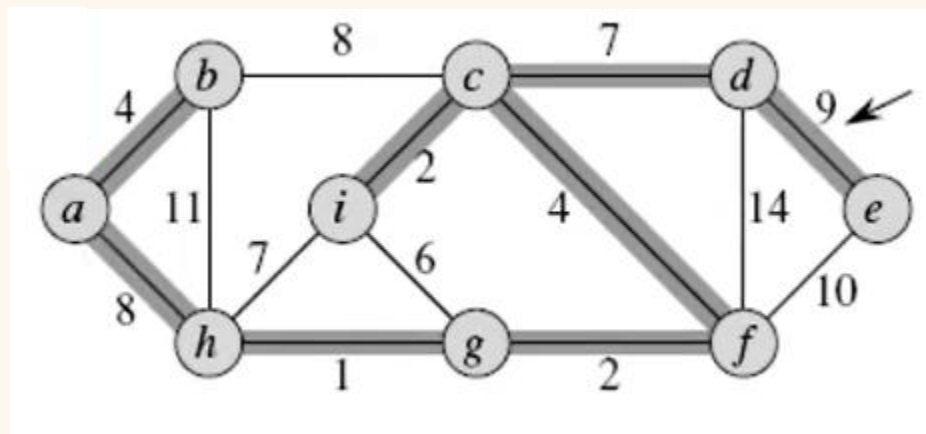
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{ {a, b, c, d, e, f, g, h, i}

e e f pertencem a
mesma árvore na
floresta?

(e,f) fecha um ciclo.



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); ~~(e, f)~~; (b, h); (d, f)

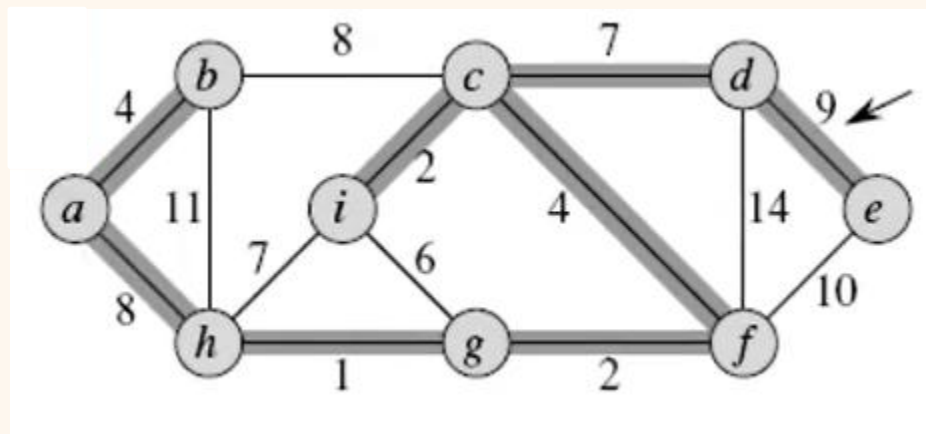
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{ {a, b, c, d, e, f, g, h, i}}

b e h pertencem a
mesma árvore na
floresta?

(b,h) fecha um ciclo.



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); ~~(e, f)~~; ~~(b, h)~~; (d, f)

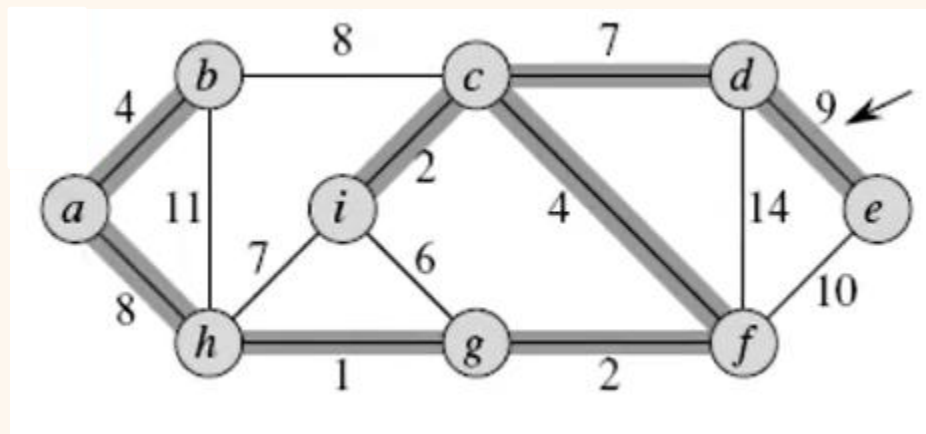
Algoritmo de Kruskal

- 3º passo: para cada aresta ordenada, faça...

{ {a, b, c, d, e, f, g, h, i}

d e f pertencem a
mesma árvore na
floresta?

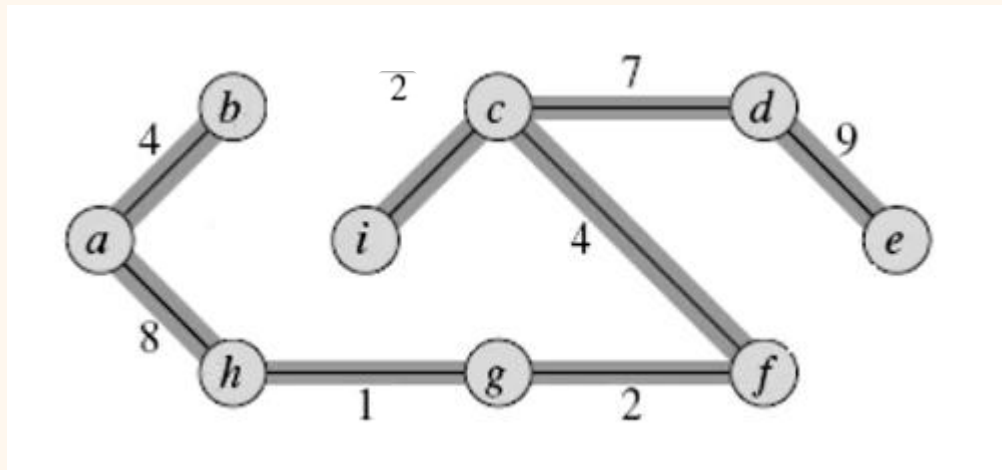
(d,f) fecha um ciclo.



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); ~~(e, f)~~; ~~(b, h)~~; ~~(d, f)~~

Algoritmo de Kruskal

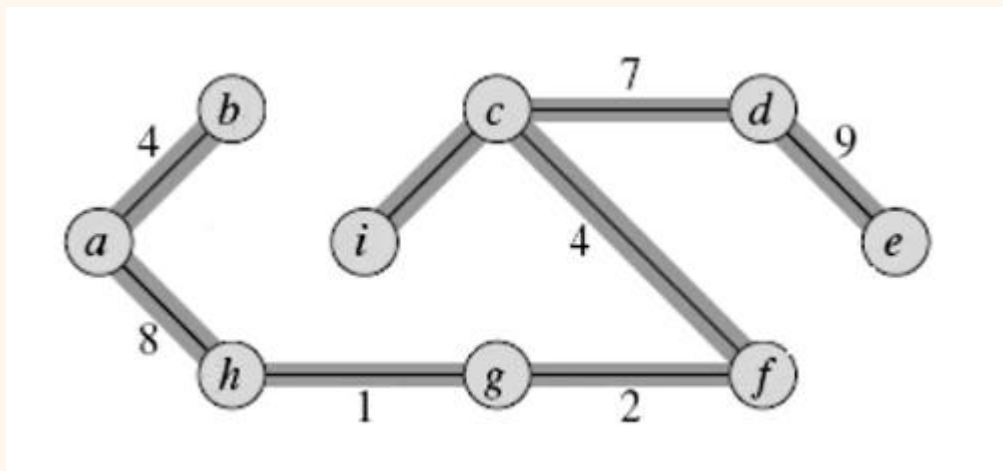
$$\text{Custo} = 4+8+2+1+2+4+7+9 = 37$$



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); ~~(e, f)~~; ~~(b, h)~~; ~~(d, f)~~

Algoritmo de Kruskal

- O conjunto X (arestas da AGM) foi composto ao longo da execução do Kruskal, onde apenas as arestas não marcadas de A' foram adicionadas à árvore.



A' (g, h); (c, i); (f, g); (a, b); (c, f); ~~(g, i)~~; (c, d); ~~(h, i)~~; (a, h); ~~(b, c)~~; (d, e); ~~(e, f)~~; ~~(b, h)~~; ~~(d, f)~~

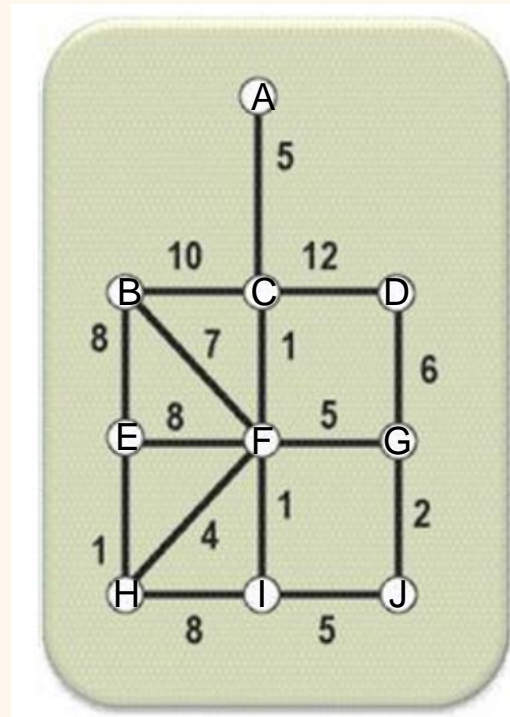
Algoritmo de Kruskal

Ponto Chave:

- Ele encontra uma aresta segura para adicionar à floresta encontrando, de todas as arestas que conectam duas árvores quaisquer, uma aresta de peso mínimo;
- Kruskal é considerado um algoritmo guloso, porque em cada passo ele adiciona à floresta uma aresta de peso mínimo (daquelas que ainda podem ser adicionadas). Ou seja, faz uma avaliação dentre todas as possibilidades que possui.

Algoritmo de Kruskal

- Encontre a árvore geradora mínima do grafo ao lado usando Kruskal.
- Qual a aresta incluída na quarta iteração do algoritmo?
- Qual o valor do custo da AGM?



Algoritmos de Kruskal x Prim

- Prim:
 - Gera uma árvore única;
 - Ao longo do algoritmo, o conjunto X sempre será uma árvore.
- Kruskal:
 - Gera uma floresta, antes de gerar a AGM;
 - Existe garantia de ser apenas uma árvore apenas depois da última iteração.

Algoritmos de Kruskal x Prim

- Questões:
 - Os algoritmos de Prim e Kruskal, quando solucionam um mesmo grafo, produzem a mesma árvore geradora mínima?
 - *Nem sempre, podem gerar árvores diferentes.*
 - Os algoritmos Prim e Kruskal, quando solucionnam um mesmo grafo, produzem árvores geradoras mínima com o mesmo custo?
 - *Sim, sempre terão o mesmo custo.*

Algoritmo Deleção-Reversa

- Reverse-delete algorithm,
- Contrário de Kruskal: Removendo arestas de maior custo

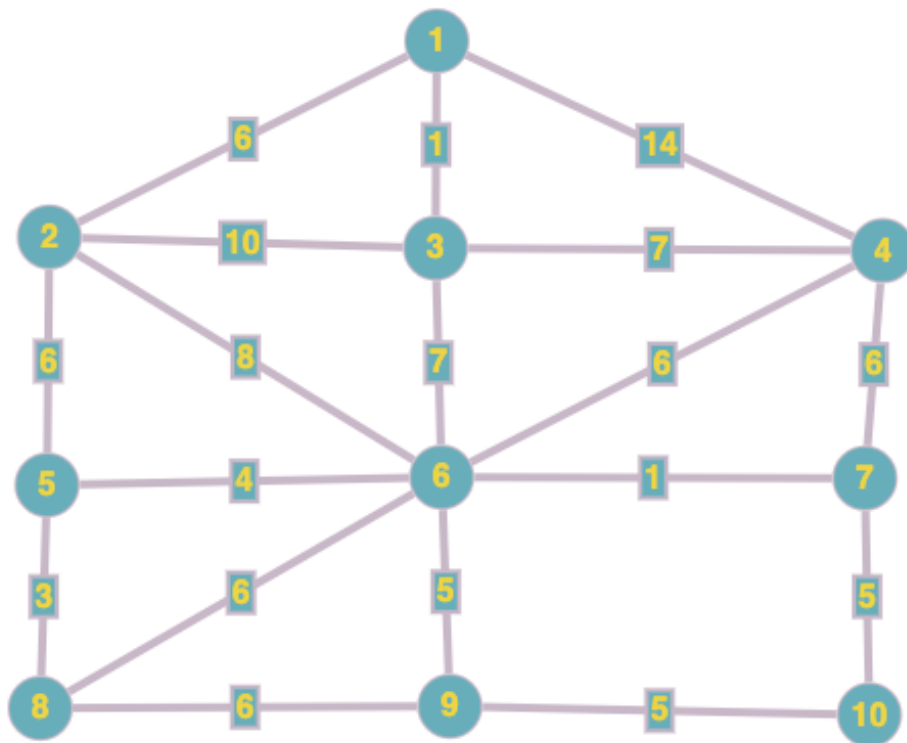
```
function ReverseDelete(edges[] E) is  
    sort E in decreasing order  
    Define an index i  $\leftarrow$  0  
  
    while i < size(E) do  
        Define edge  $\leftarrow$  E[i]  
        delete E[i]  
        if graph is not connected then  
            E[i]  $\leftarrow$  edge  
            i  $\leftarrow$  i + 1  
  
    return edges[] E
```

Algoritmo Deleção-Reversa

- Reverse-delete algorithm:
 - Start with graph G , which contains a list of edges E .
 - Go through E in decreasing order of edge weights.
 - For each edge, check if deleting the edge will further disconnect the graph.
 - Perform any deletion that does not lead to additional disconnection.

Algoritmos de Kruskal x Prim

- Encontre a árvore geradora mínima do grafo ao lado usando Prim (a partir do vértice 1) e Kruskal.
- Qual o custo da árvore geradora mínima encontrado em Prim e em Kruskal?



Dúvidas?



Laura Alves Pacífico
laps@cesar.school
Slack: Laura Pacífico