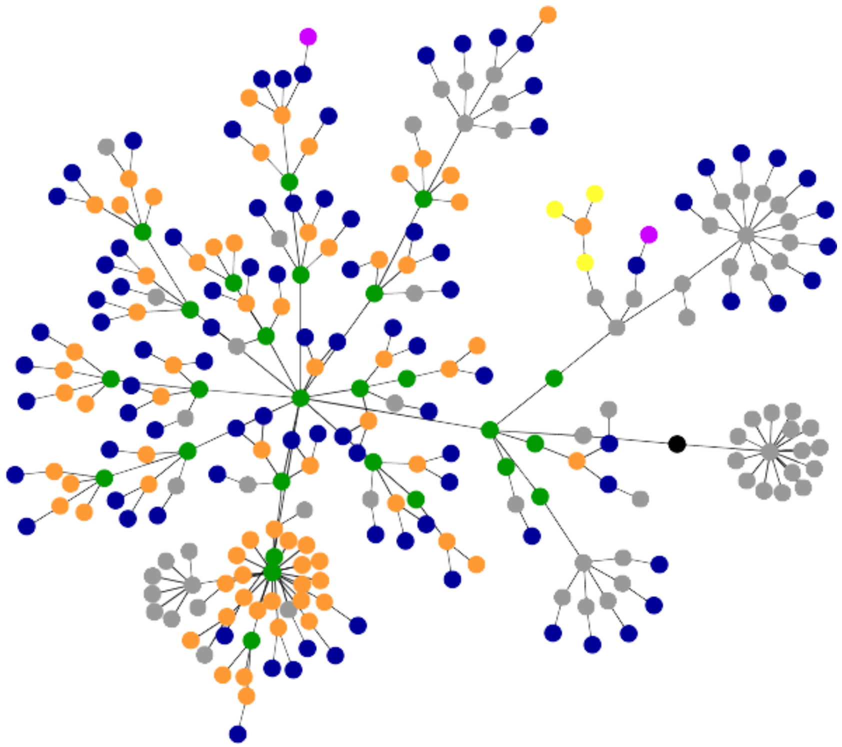
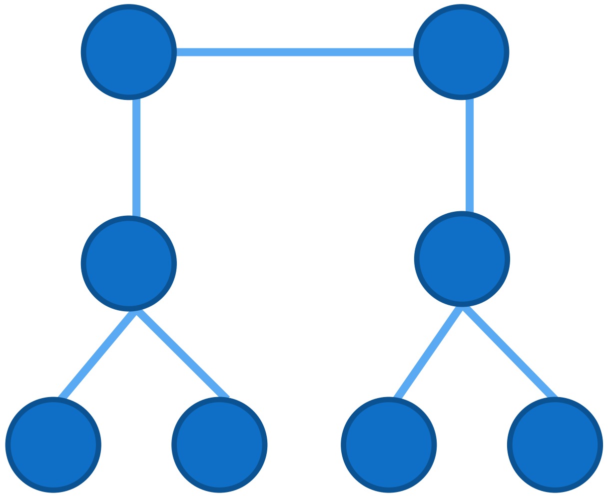


Árvores

Continuação…

Árvore

Grafo conexo e acíclico

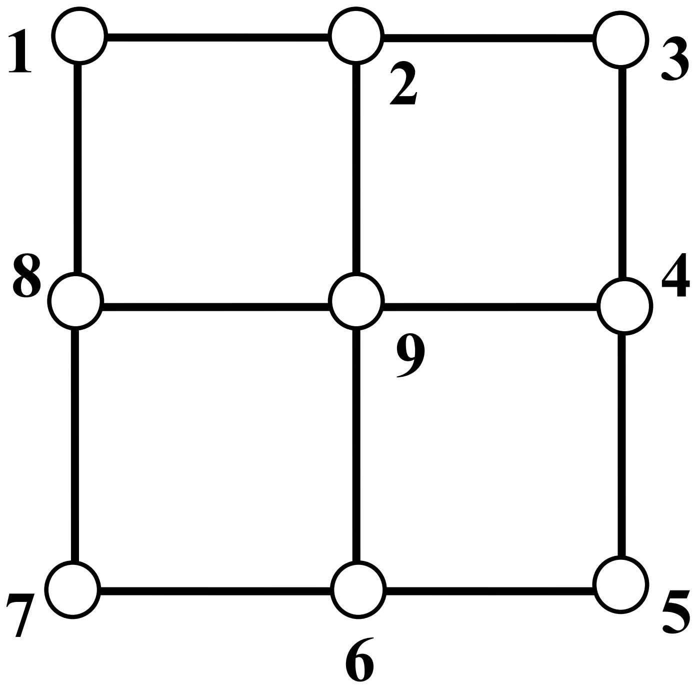


Teorema de Caracterização de Árvores

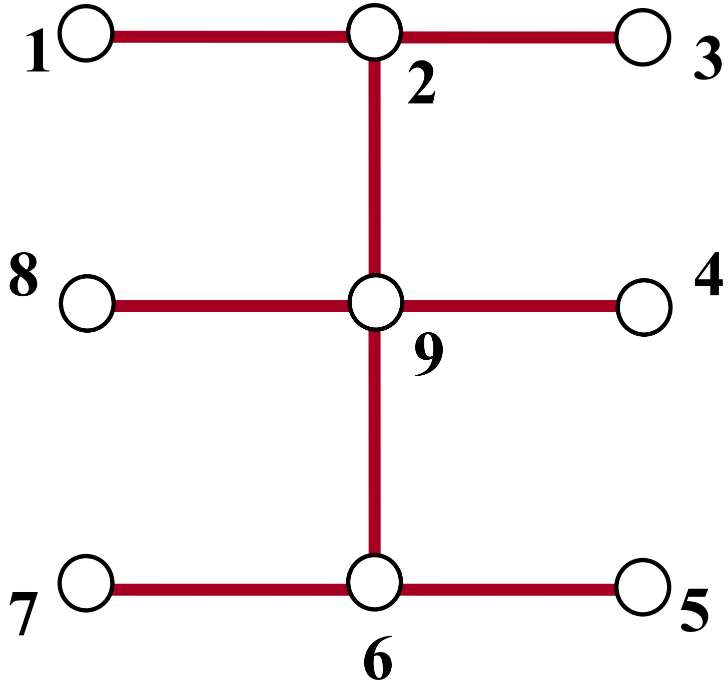
1. *G* é uma árvore
2. Existe exatamente um caminho entre cada par de vértices de *G*
3. *G* é conexo e m=n-1 (m=número de arestas, n=número de vértices)
4. *G* é acíclico e m=n-1
5. *G* é acíclico e se quaisquer dois vértices não adjacentes de *G* forem conectados por uma aresta, então o grafo resultante conterá exatamente um ciclo.

Árvore Geradora

Uma ***Árvore Geradora*** de um grafo $G = (V,E)$ é um subgrafo gerador $T=(V,E_T)$ conexo e acíclico.



G



T

Árvore Geradora Mínima

Como Prim (1957) pensou?

Escolhe um vértice qualquer.
Inclui a menor aresta incidente a um vértice da árvore, tal que o outro terminal da aresta não esteja na árvore.
Parar quando todos os vértices estiverem na árvore.

Prim, R. C. Shortest connection networks and some generalizations. *Bell System Technical Journal*, v. 36, pp. 1389-1401, 1957.

Algoritmo de Prim

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

Escolha qualquer vértice $i \in N$

$Z \leftarrow \{i\}$

$V \leftarrow N \setminus \{i\}$

$T_{min} \leftarrow \emptyset$

Enquanto $Z \neq N$ **Faça**

  Encontrar a aresta $(j,k) \in M$ tal que $j \in Z$, $k \in V$
  e d_{jk} é mínimo

    $Z \leftarrow Z \cup \{k\}$

    $V \leftarrow V \setminus \{k\}$

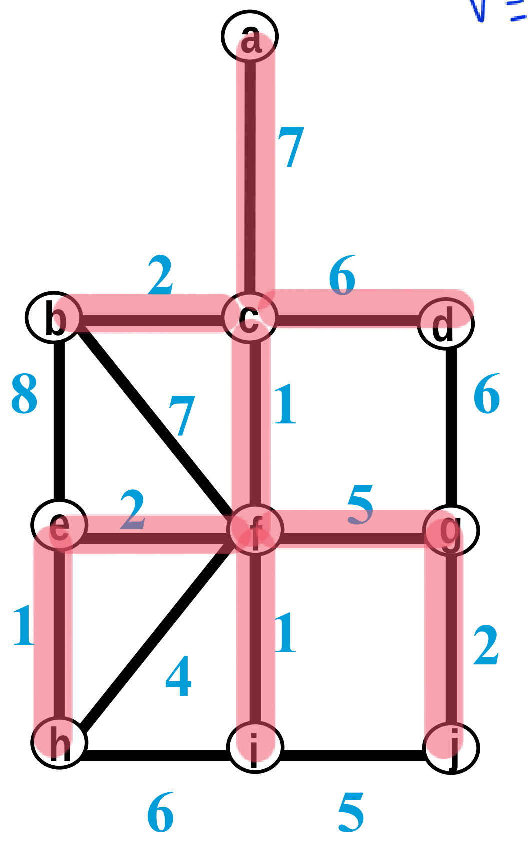
    $\boldsymbol{T_{Min}} \leftarrow \boldsymbol{T_{Min}} \cup (j,k)$

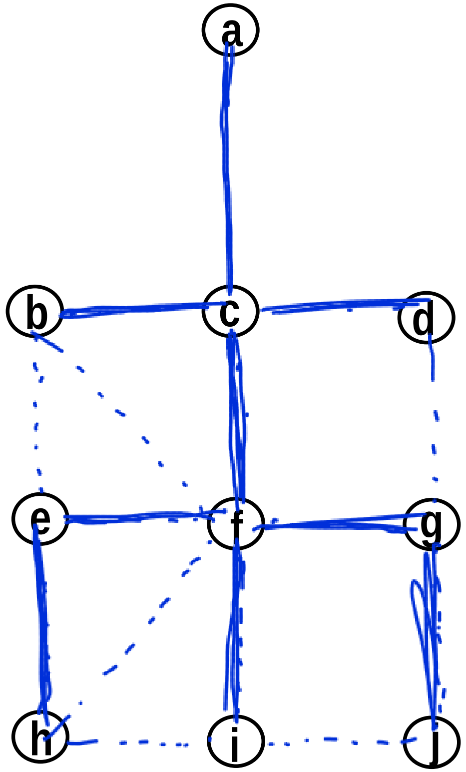
Escrever $\boldsymbol{T_{Min}}$ {o conjunto das arestas da árvore geradora mínima}

Algoritmo de Prim

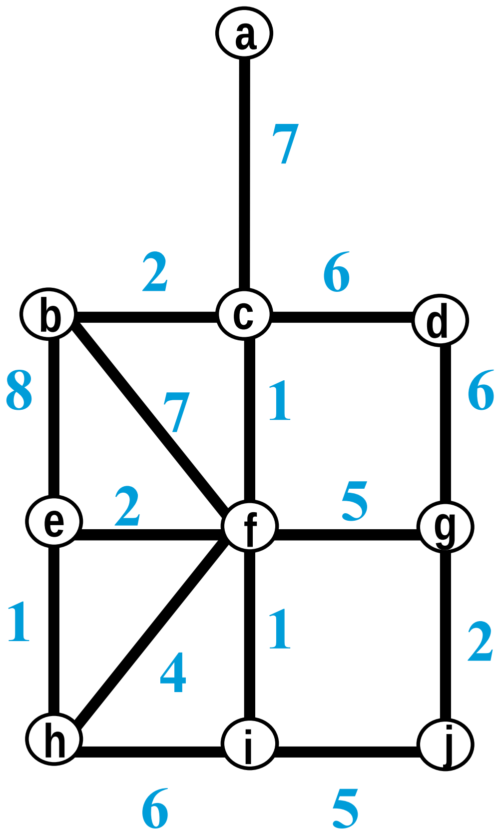
$Z = \{a, c, f, i, b, e, h, g, j, d\}$

$V = \{\ \}$



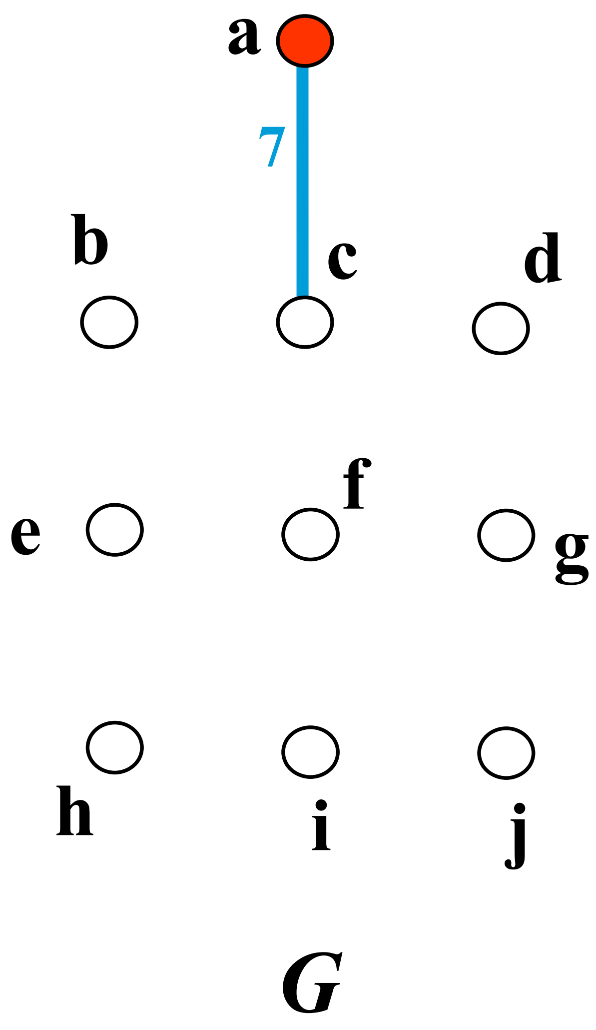


Algoritmo de Prim



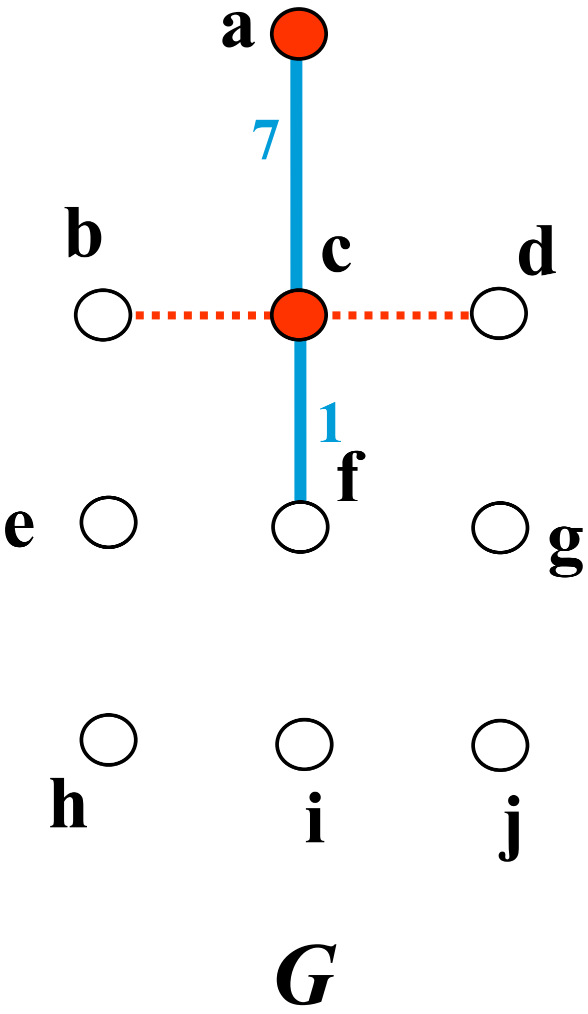
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



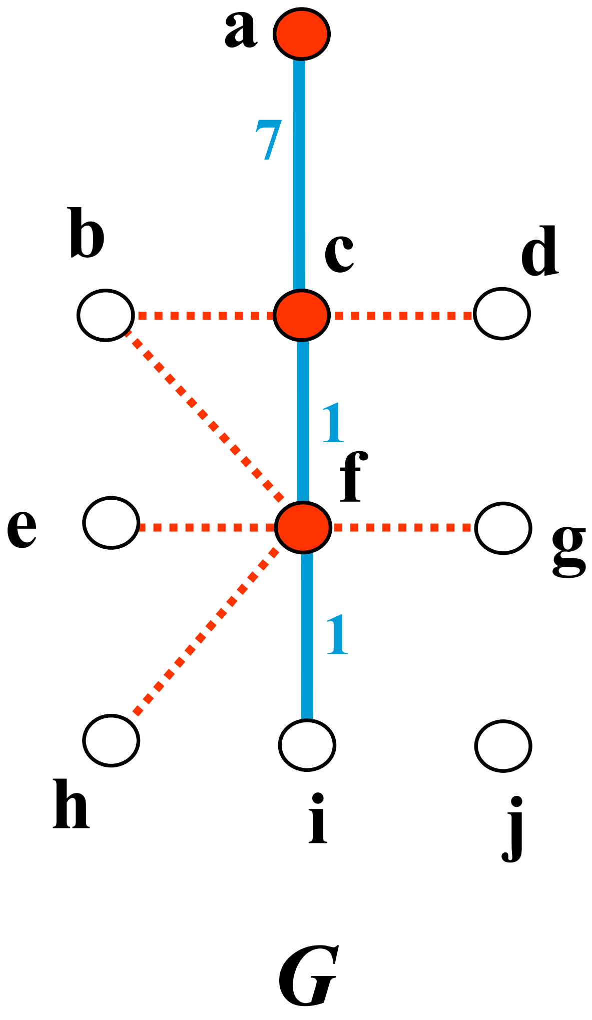
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



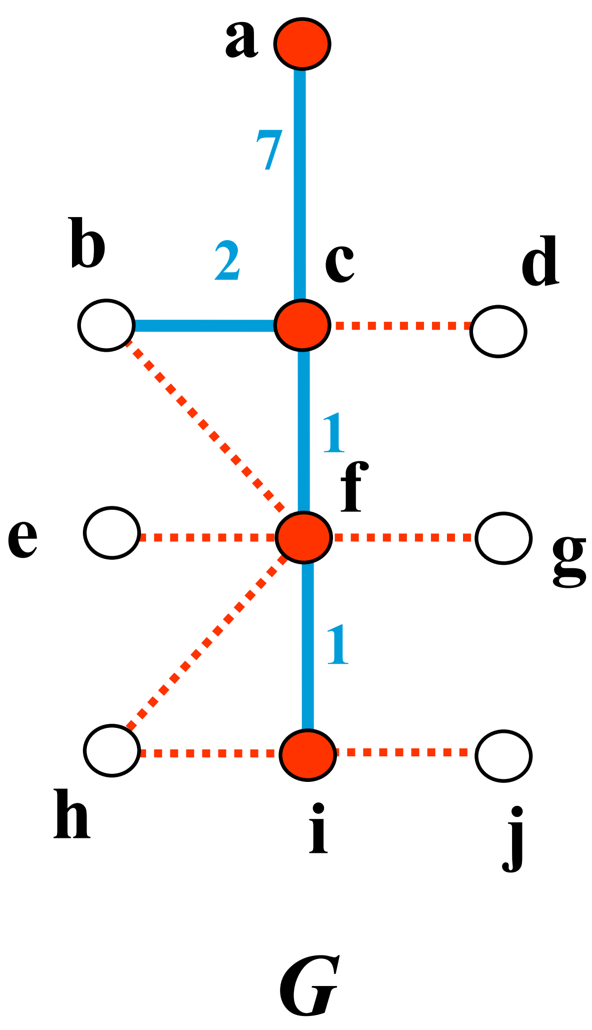
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



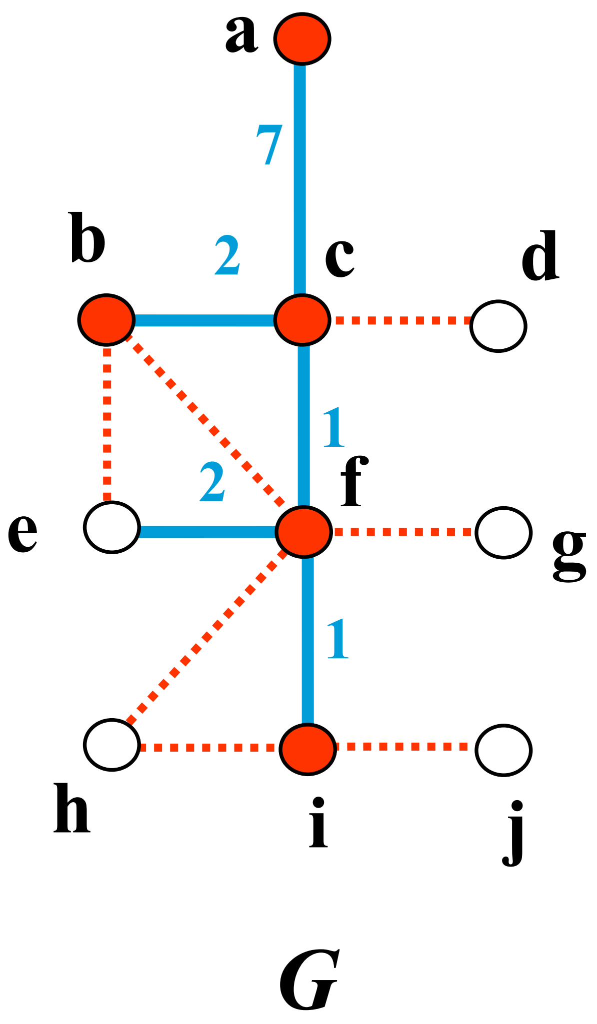
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



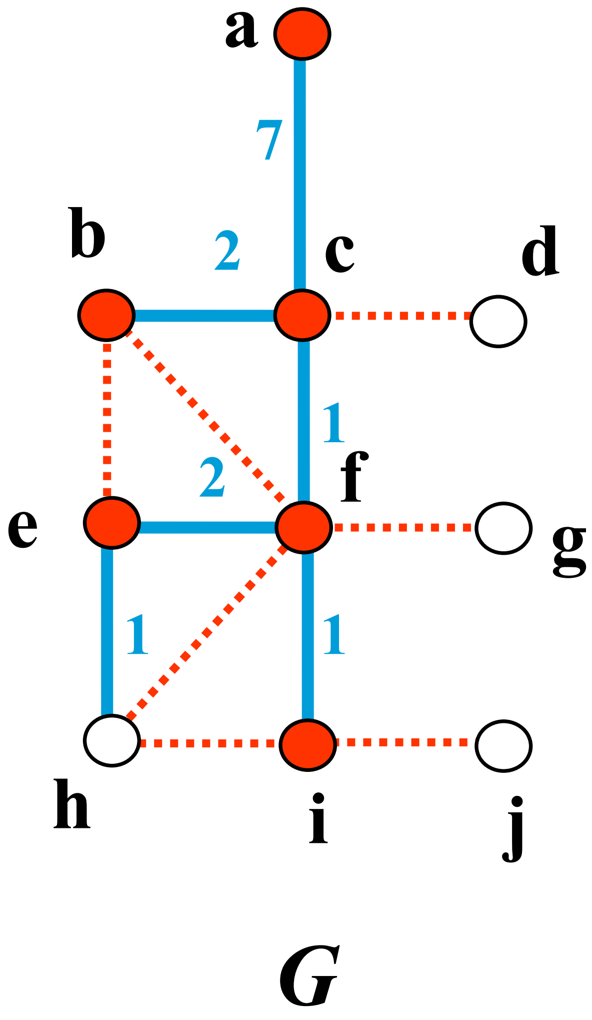
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



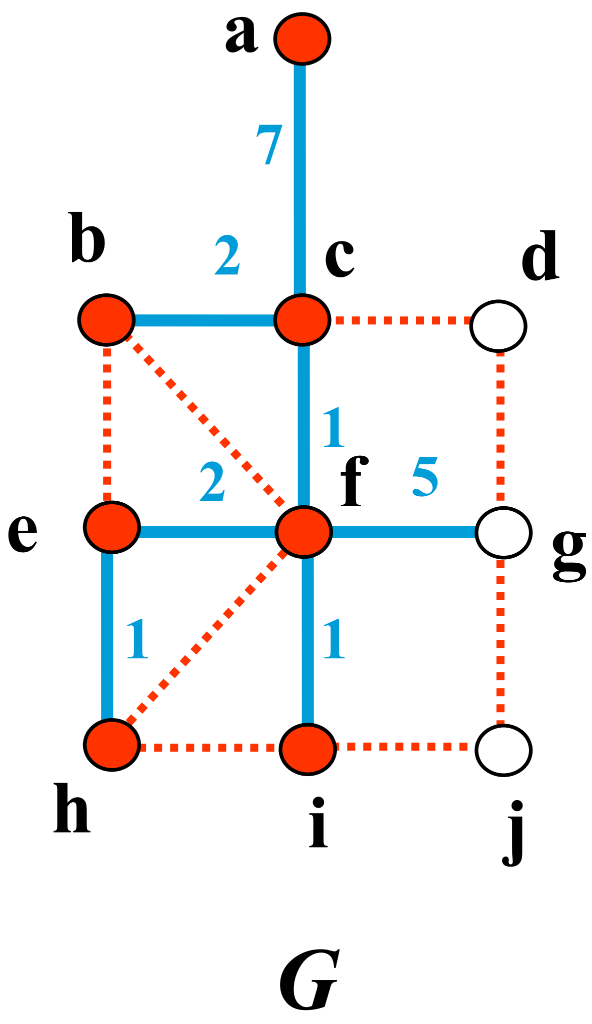
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



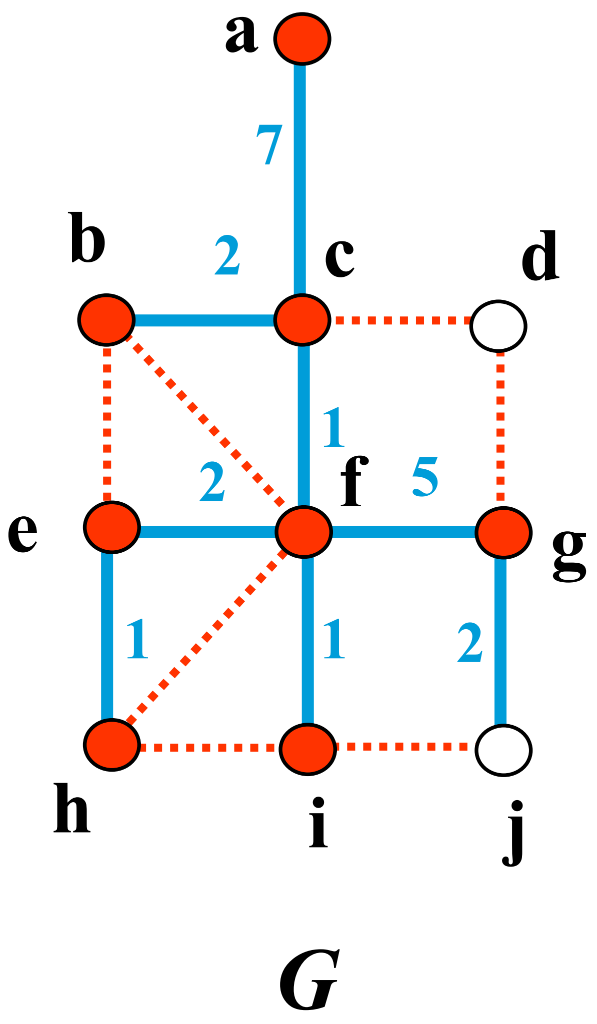
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



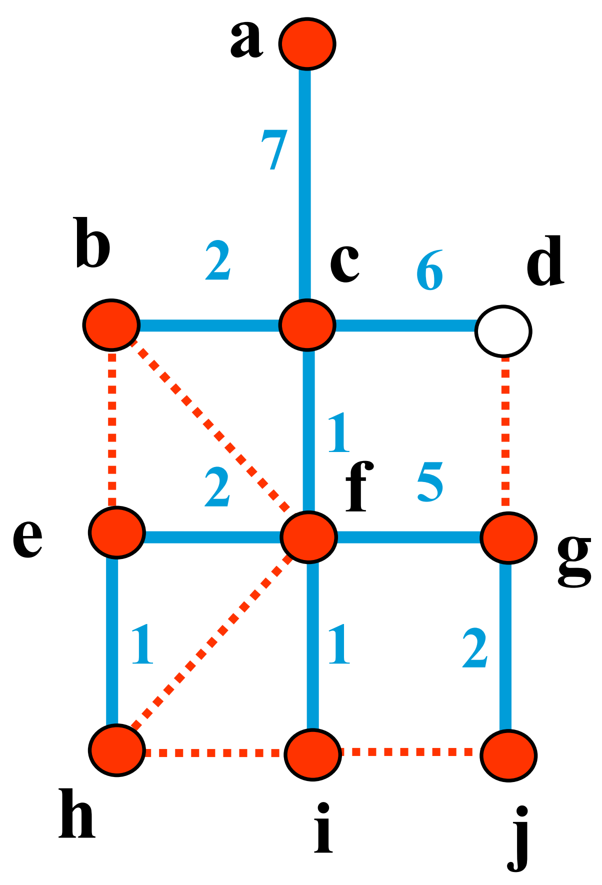
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

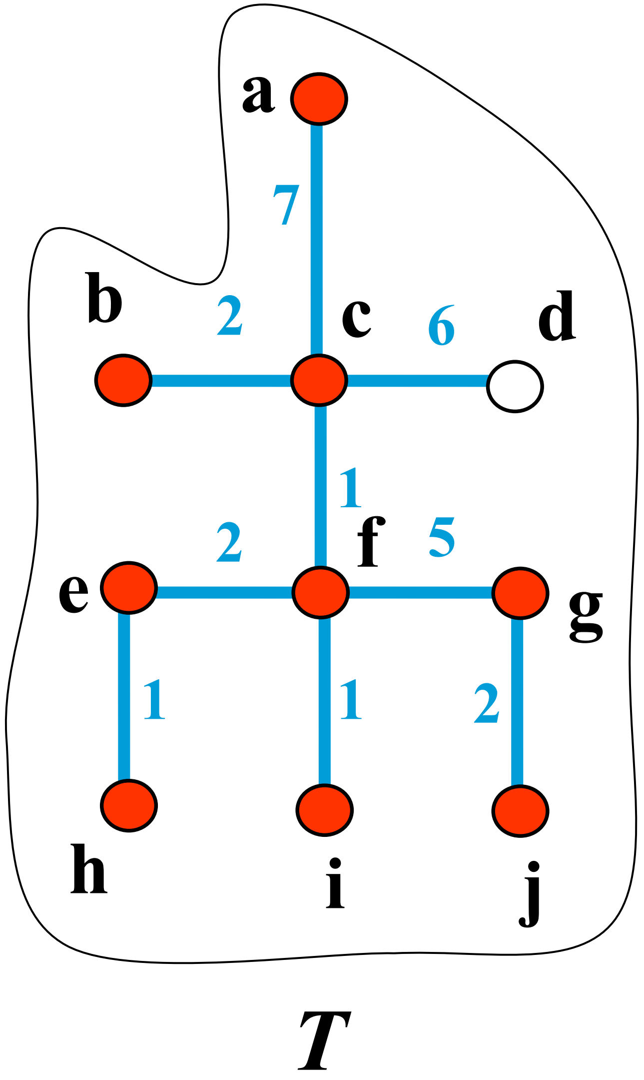
Algoritmo de Prim



G

	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

Algoritmo de Prim



Custo = 27

Algoritmo de Prim

Exercício

O algoritmo de Prim produz uma árvore geradora?

Neste caso, ela é mínima?

Ler G=(N,M) e D=[d_{ij}] a matriz distância de G
Escolha qualquer vértice $i \in N$
$Z \leftarrow \{i\}$
$V \leftarrow N \setminus \{i\}$
$T_{min} \leftarrow \varnothing$
Enquanto $Z \neq N$ **Faça**
  Encontrar a aresta (j,k) $\in M$ tal que $j \in Z$, $k \in V$ e d_{jk} é mínimo
    $Z \leftarrow Z \cup \{k\}$
    $V \leftarrow V \setminus \{k\}$
    $T_{Min} \leftarrow T_{Min} \cup (j,k)$
Escrever T_{Min}

Algoritmo de Prim

Complexidade

(Prova – Cormen et al., 2002)

O "enquanto" faz *n*-1 iterações, *O*(*n*).

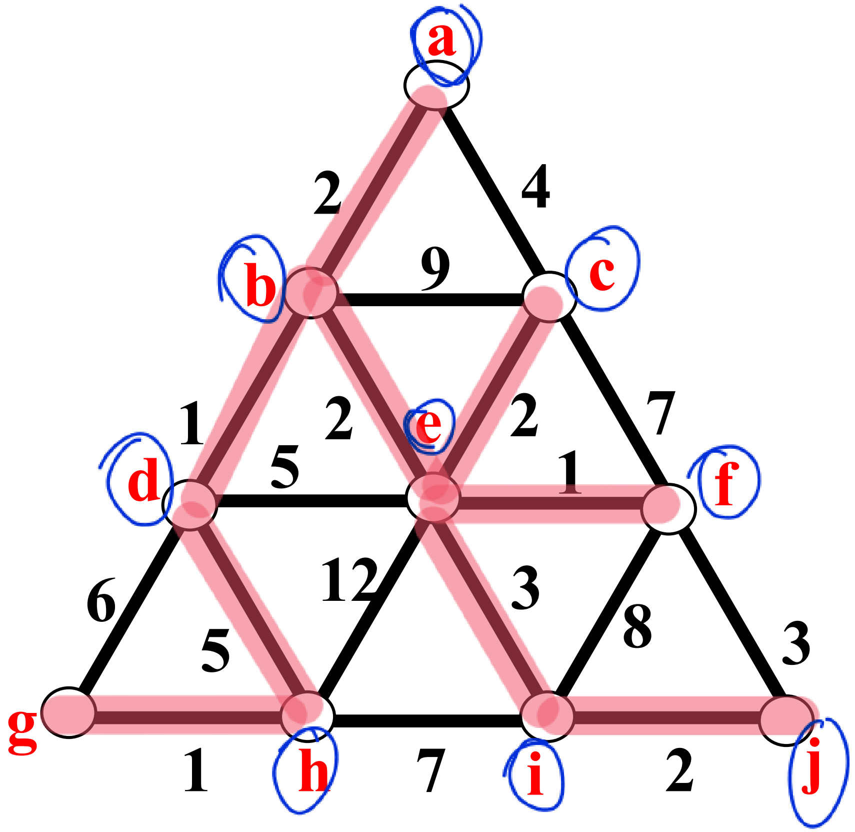
Encontrar a aresta $(j,k) \in M$ tal que $j \in Z$, $k \in V$ e d_{jk} é mínimo

Complexidade: (?)

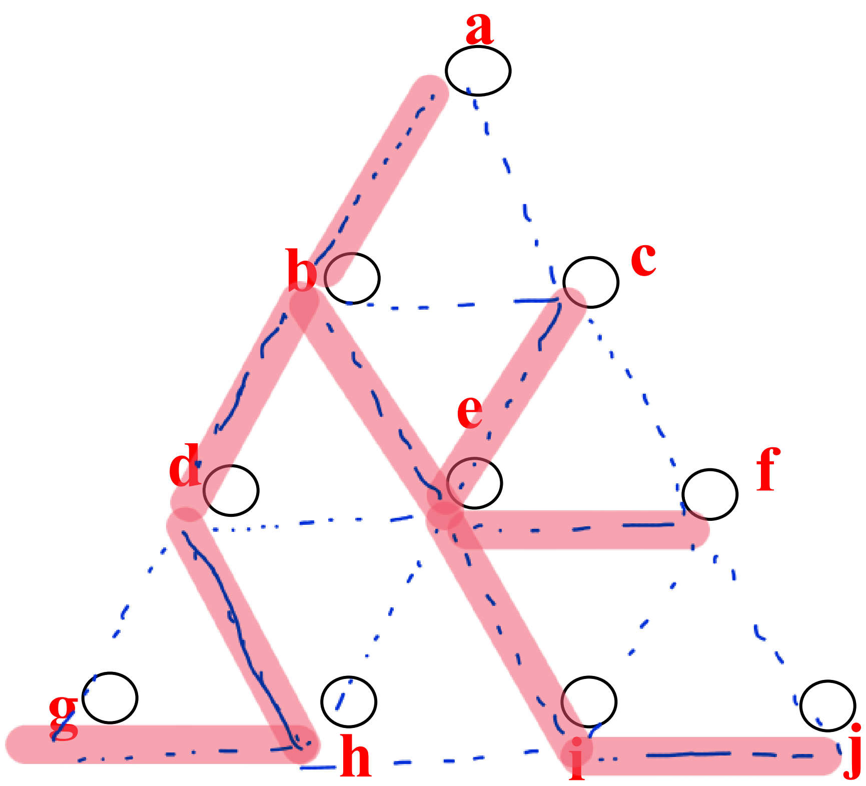
Ler $G=(N,M)$ e $D=[d_{ij}]$ a matriz distância de G
Escolha qualquer vértice $i \in N$
$Z \leftarrow \{i\}$
$V \leftarrow N \setminus \{i\}$
$T_{min} \leftarrow \varnothing$
Enquanto $Z \neq N$ **Faça**
  Encontrar a aresta $(j,k) \in M$ tal que $j \in Z$, $k \in V$ e d_{jk} é mínimo
    $Z \leftarrow Z \cup \{k\}$
    $V \leftarrow V \setminus \{k\}$
    $\boldsymbol{T_{Min}} \leftarrow \boldsymbol{T_{Min}} \cup (j,k)$
Escrever $\boldsymbol{T_{Min}}$

Algoritmo de Prim

Exercício Prim



G



Árvore Geradora Mínima

Como Borůvka (1926) pensou?

Fonte: Wikipedia

Forma uma floresta com cada vértice sendo uma árvore.
Liga cada árvore com outra pela aresta de peso mínimo.
Parar quando for uma árvore.

Borůvka, Otakar (1926), "O jistém problému minimálním", Práce Moravské přírodovědecké společnosti, ***3*** *(3): 37–58*

Algoritmo de Borůvka

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

F_0 é uma floresta inicial com n subárvores T_j , j=1,..,n

$i \leftarrow 0$

Enquanto F_i não for uma árvore **Faça**

Para cada $T_j \in F_i$

Determine a menor aresta (x_α, y_α) incidente em T_j onde $x_\alpha \in T_j$ e $y_\alpha \notin T_j$

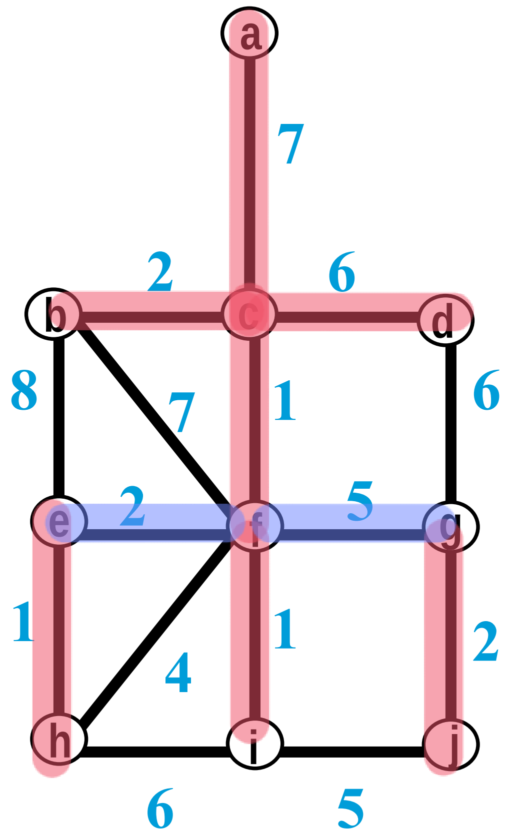
$$F_{i+1} \leftarrow F_i \cup \left[\bigcup_\alpha (x_\alpha, y_\alpha) \right]$$

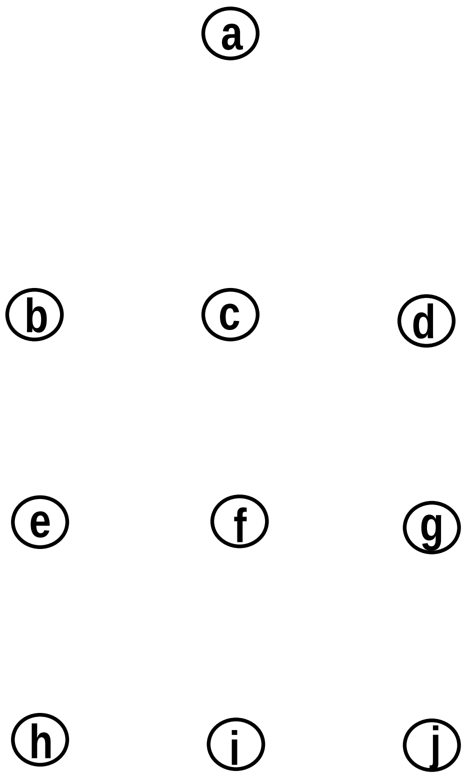
Faça

$i \leftarrow i+1$

Escrever F_i {arestas da árvore geradora mínima}

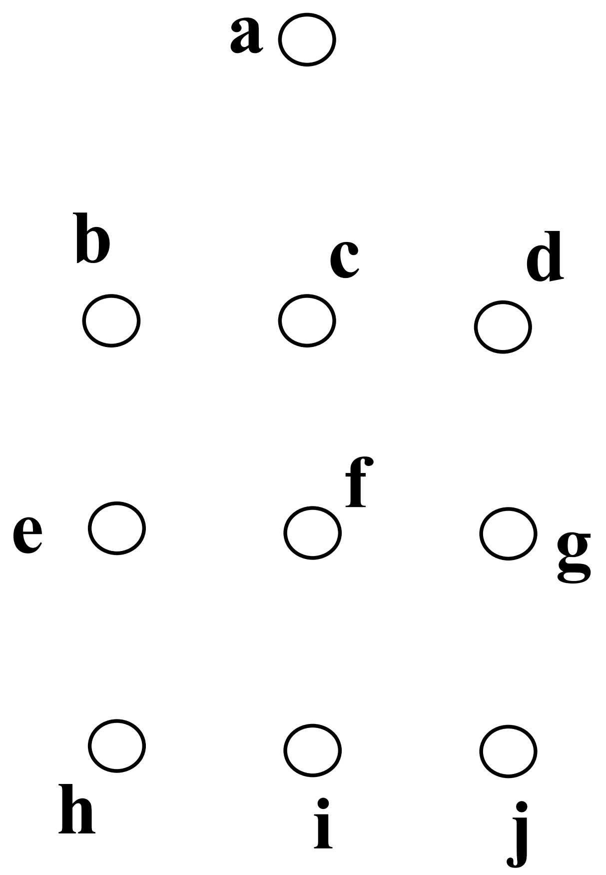
Algoritmo de Borůvka





Algoritmo de Borůvka

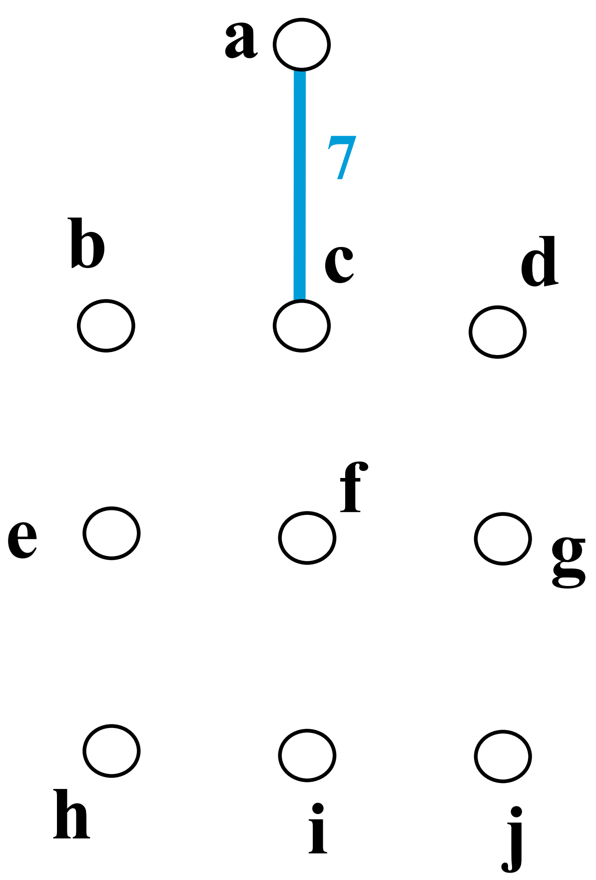
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



F

Algoritmo de Borůvka

	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

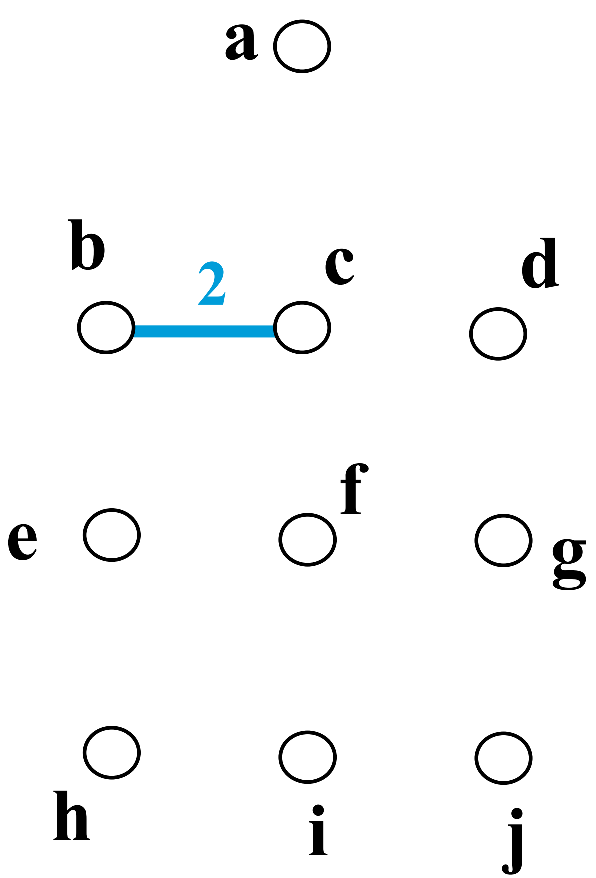


Vértice a

F_a

Algoritmo de Borůvka

	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞

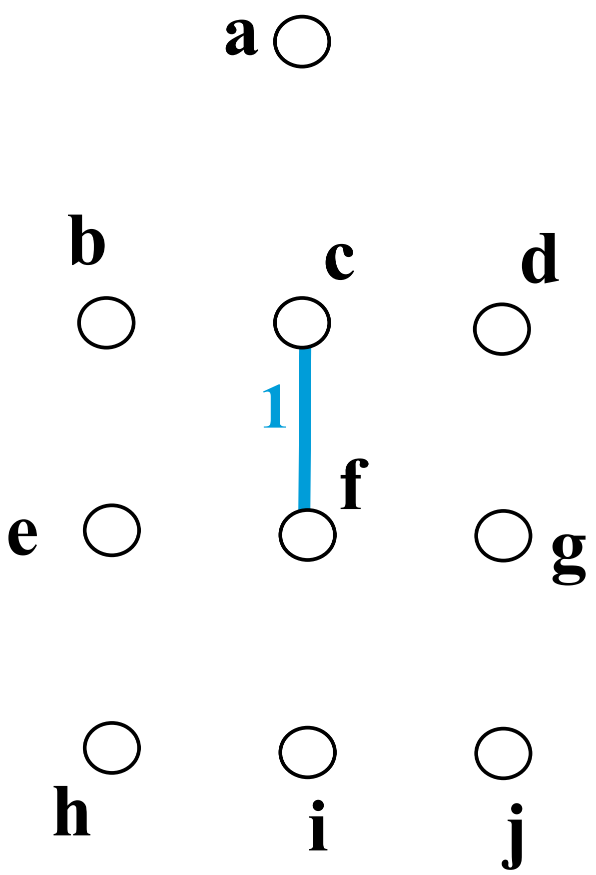


Vértice b

F_b

Algoritmo de Borůvka

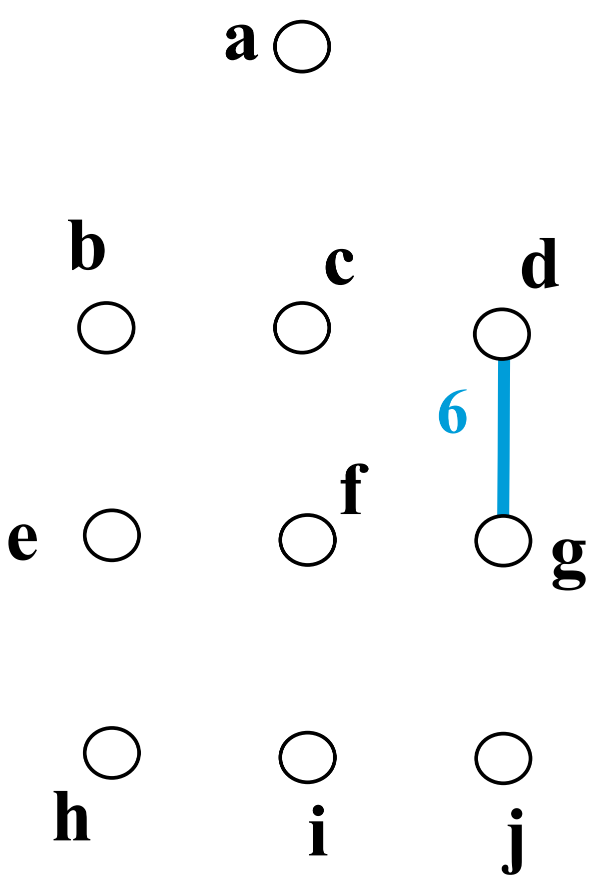
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice c

Algoritmo de Borůvka

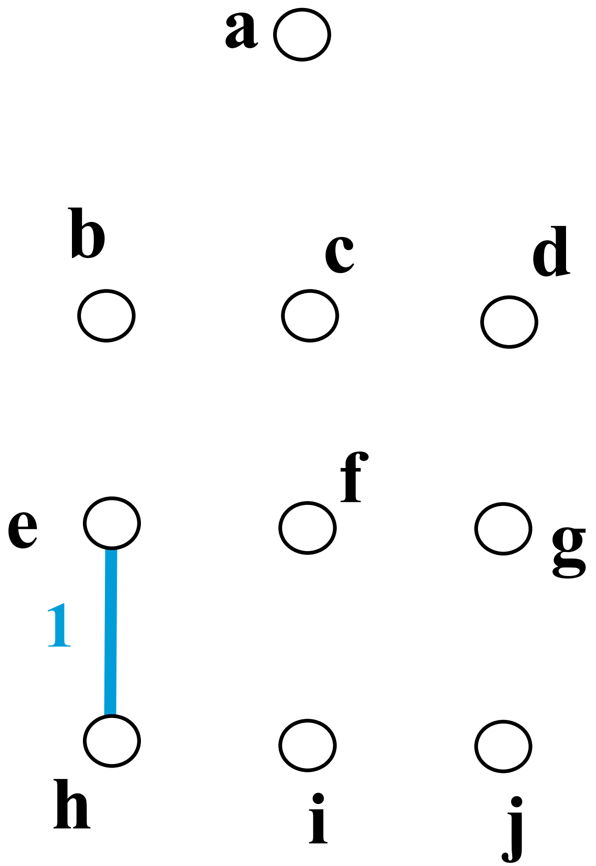
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice d

Algoritmo de Borůvka

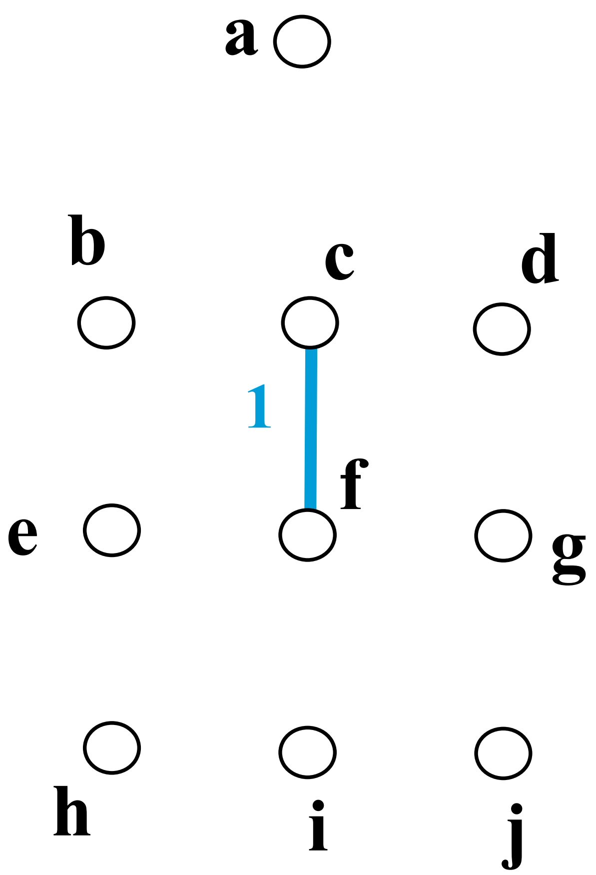
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice e

Algoritmo de Borůvka

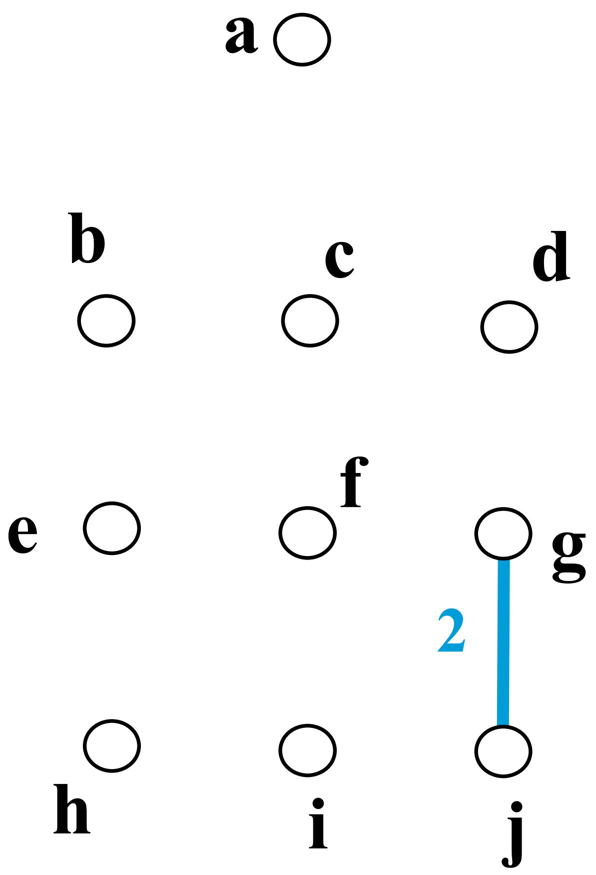
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice f

Algoritmo de Borůvka

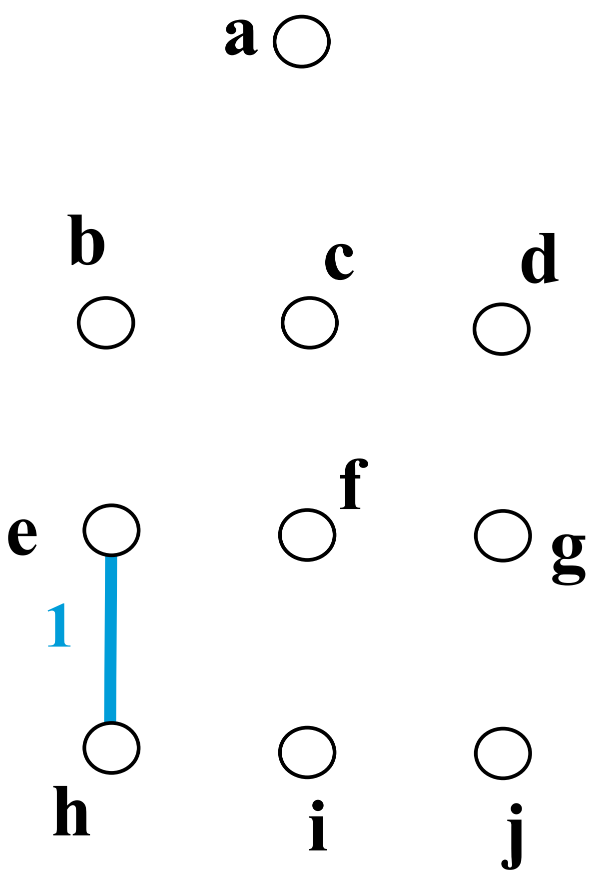
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice g

Algoritmo de Borůvka

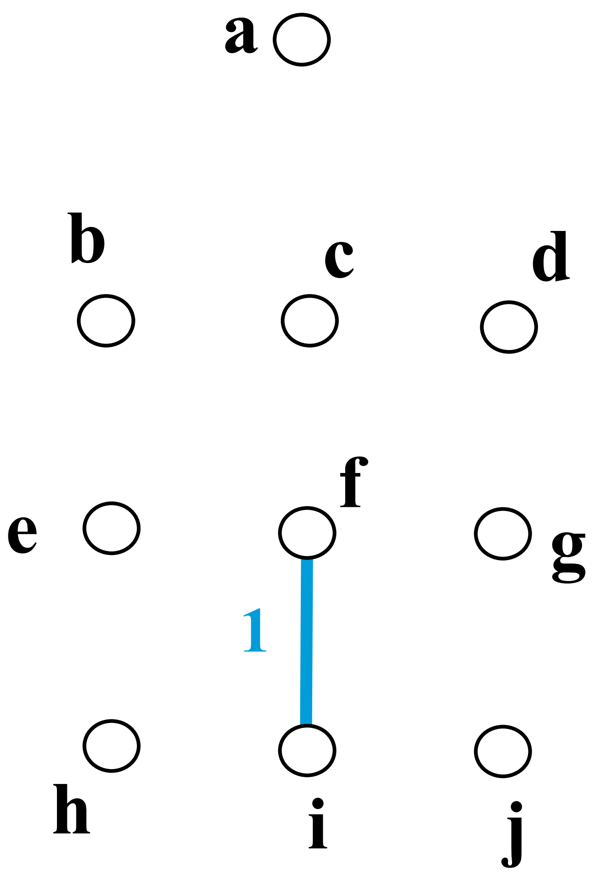
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice h

Algoritmo de Borůvka

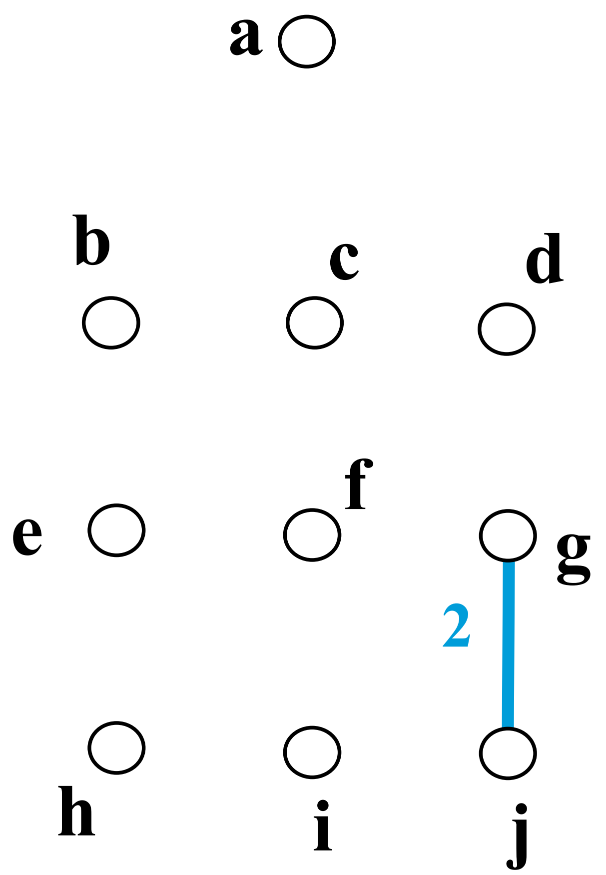
	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



Vértice i

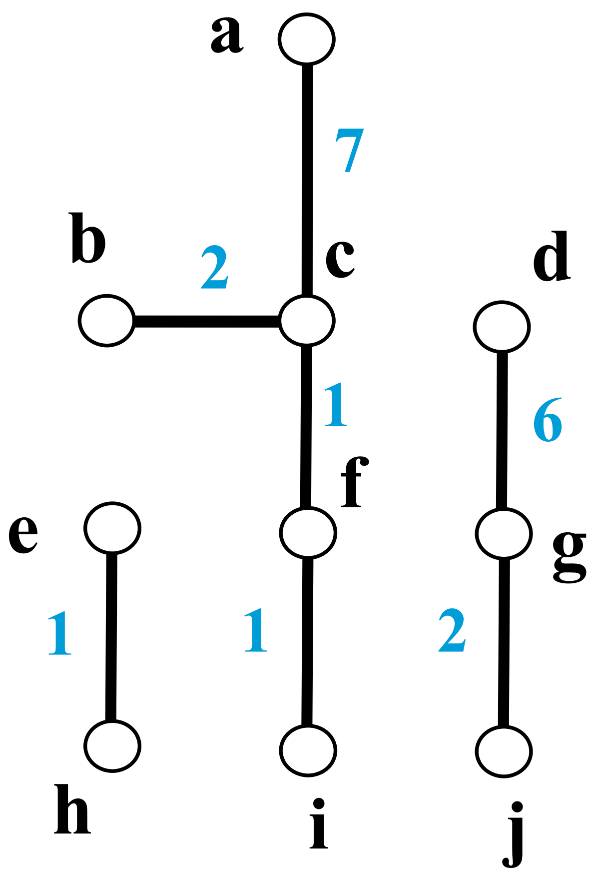
Algoritmo de Borůvka

	a	b	c	d	e	f	g	h	i	j
a	∞	∞	7	∞	∞	∞	∞	∞	∞	∞
b	∞	∞	2	∞	8	7	∞	∞	∞	∞
c	7	2	∞	6	∞	1	∞	∞	∞	∞
d	∞	∞	6	∞	∞	∞	6	∞	∞	∞
e	∞	8	∞	∞	∞	2	∞	1	∞	∞
f	∞	7	1	∞	2	∞	5	4	1	∞
g	∞	∞	∞	6	∞	5	∞	∞	∞	2
h	∞	∞	∞	∞	1	4	∞	∞	6	∞
i	∞	∞	∞	∞	∞	1	∞	6	∞	5
j	∞	∞	∞	∞	∞	∞	2	∞	5	∞



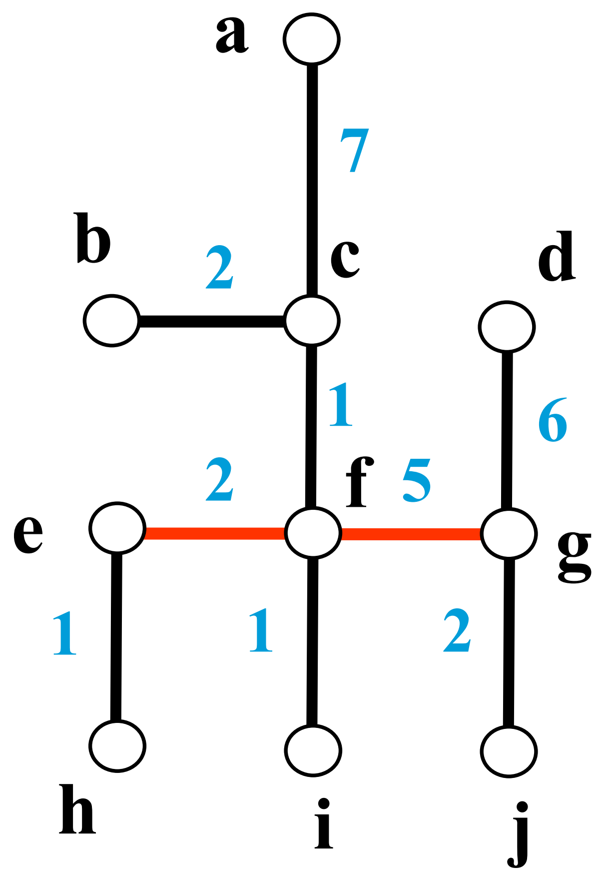
Vértice j

Algoritmo de Borůvka



Floresta da Primeira Iteração

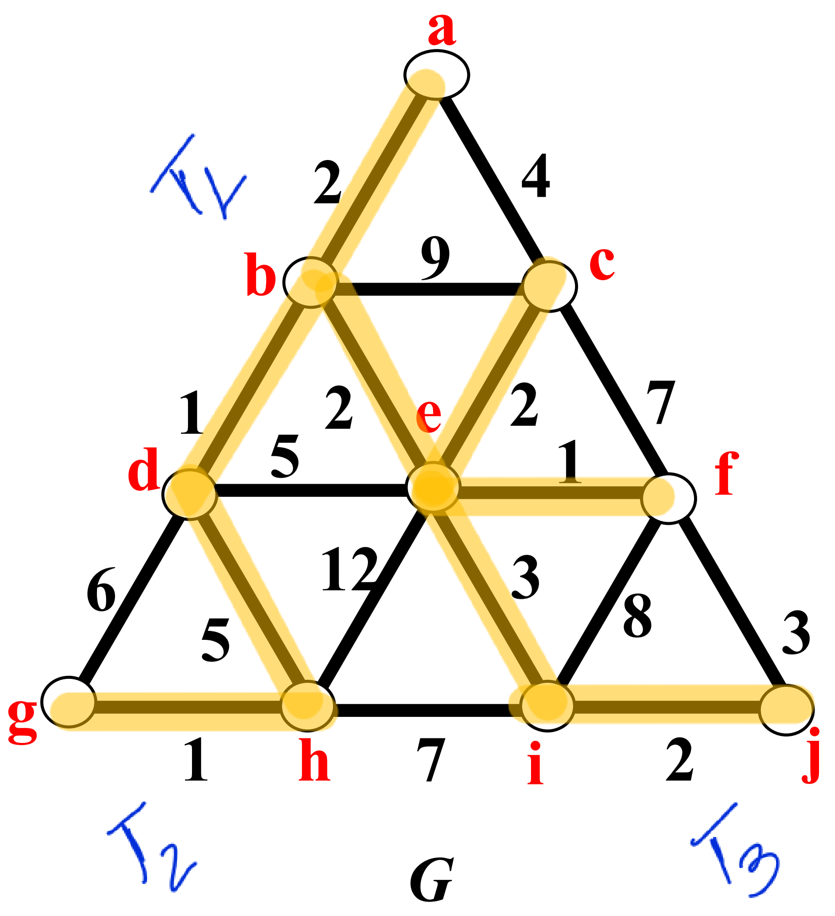
Algoritmo de Borůvka

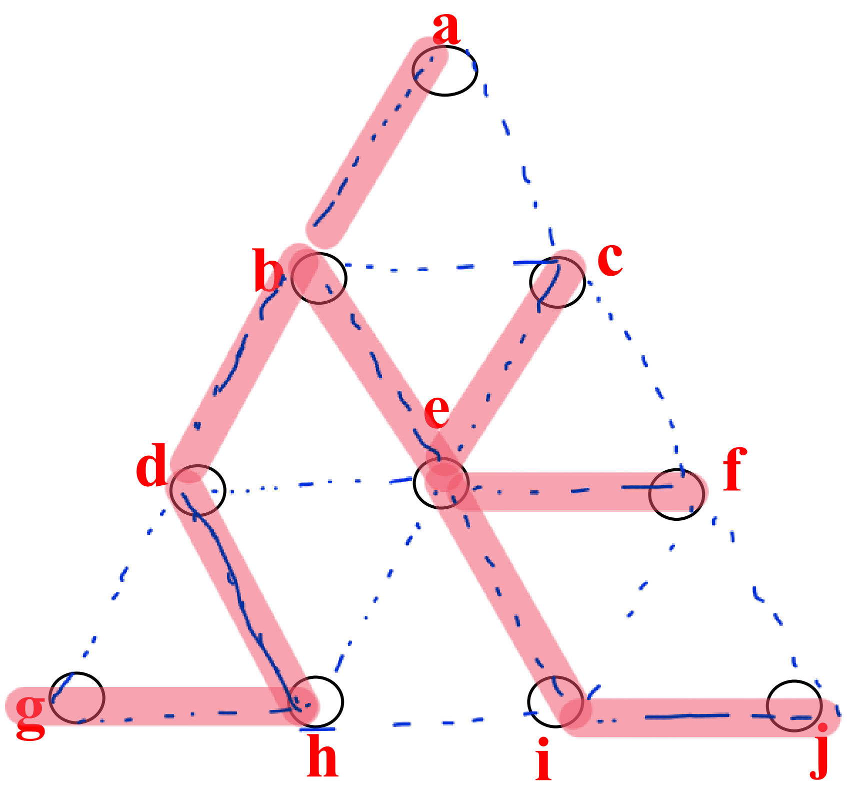


Arestas da segunda iteração

Algoritmo de Borůvka

Exercício Borůvka





Algoritmo de Borůvka

Como cada árvore da floresta inclui a menor aresta incidente na solução em cada iteração, se existirem arestas iguais no grafo pode ocorrer **ERRO** formando ciclo.

O algoritmo, da forma descrita só funciona corretamente se as arestas de *G* forem distintas. Para evitar que possam ocorrer inclusões indevidas por ocasião das uniões das subárvores deve-se ordenar lexicograficamente as arestas de *G*.

Exercício: Exemplificar um caso onde a aplicação do algoritmo sem a ordenação lexicográfica FALHA.

Algoritmo de Borůvka

Em cada iteração do comando enquanto, o número de árvores da floresta diminui de um fator de 2.

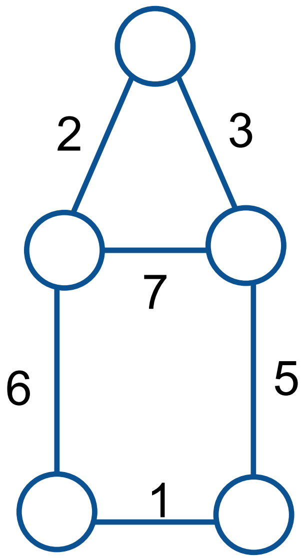
Portanto, são executadas $\log_2 n$ iterações do laço mais externo.

Em cada iteração são examinadas $O(m)$ arestas. Portanto, o algoritmo, sem estruturas de dados especiais pode ser implementado em $O(m \log n)$.

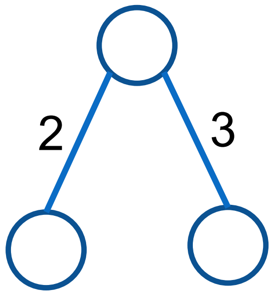
k- Árvore Geradora Mínima

Considere $G = (V,E)$, $|V| = n$, um grafo conexo no qual cada aresta e possui peso $w(e)$ e k uma constante, tal que $k < n$.

O problema consiste em encontrar a árvore geradora mínima de G com k vértices.



G



3-AGM

Ravi, R., Sundaram, R., Marathe, M. V., Rosenkrantz, D. J., & Ravi, S. S. (1996). Spanning trees—short or small. *SIAM Journal on Discrete Mathematics*, *9*(2), 178-200.

Árvores Geradoras Mínimas

Trabalho com revisão de algoritmos para árvores geradoras mínimas

- BAZLAMAÇCI, C. F.; HINDI, K. S. Minimum-weight spanning tree algorithms: a survey and empirical study. *Computers & Operations Research*, v. 28, pp. 767-785, 2001.

Referências

- CHU, Y.J.; LIU, T.H. On the shortest arborescence of a directed graph, *Science Sinica*, v.14, 1965, pp.1396-1400.
- EDMONDS, J. Optimum branchings, *J. Research of the National Bureau of Standards*, 71B, 1967, pp.233-240.
- KRUSKAL Jr, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. In: *Proceedings of the American Mathematical Society*, v. 7, n. 1, pp. 48–50, 1956.
- PRIM, R. C. Shortest connection networks and some generalizations*. Bell System Technical Journal*, v. 36, pp. 1389-1401, 1957.
- RAIDL, G. R.; JULSTROM, B. A. Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 3, pp. 225-239, 2003.
- TARJAN, R.E. Finding Optimum Branchings, *Networks*, v.7, 1977, pp.25-35.