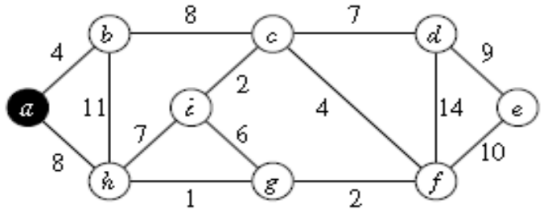
Algoritmo de Prim para árvores geradoras mínimas
Referência: Cormen, Rivest, Leiserson: Introduction to Algorithms

```
MST-PRIM(G, w, r)
 1  for each u ∈ V[G]
 2       do key[u] ← ∞
 3          π[u] ← NIL
 4  key[r] ← 0
 5  Q ← V[G]
 6  while Q ≠ ∅
 7       do u ← EXTRACT-MIN(Q)
 8          for each v ∈ Adj[u]
 9               do if v ∈ Q and w(u, v) < key[v]
10                    then π[v] ← u
11                         key[v] ← w(u, v)
```

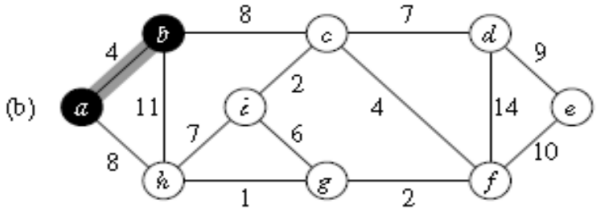
Exemplo:
Tabela abaixo apresenta a configuração das variáveis ao final de cada iteração (linha 11); 3a linha da tabela (Q) indica se o vértice está ou não na lista de prioridade (1=vértice na lista); células em vermelho correspondem aos vértices cuja distância foi alterada na iteração corrente; célula marcada com fundo amarelo indica o vértice a ser selecionado na próxima iteração

(a)



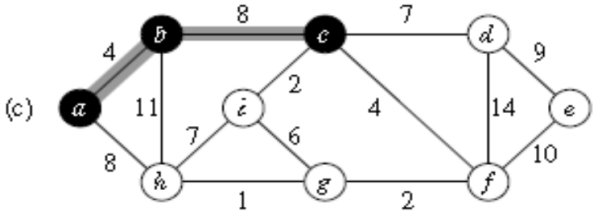
	a	b	c	d	e	f	g	h	i
π	*	a	*	*	*	*	*	a	*
key	0	4	∞	∞	∞	∞	∞	8	∞
Q		1	1	1	1	1	1	1	1

(b)



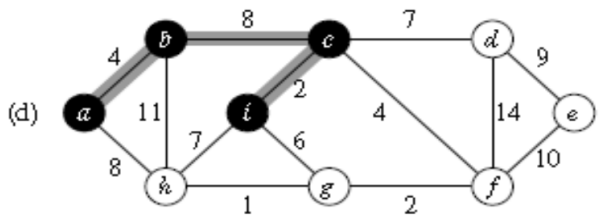
	a	b	c	d	e	f	g	h	i
π	*	a	b	*	*	*	*	a	*
key	0	4	8	∞	∞	∞	∞	8	∞
Q			1	1	1	1	1	1	1

(c)



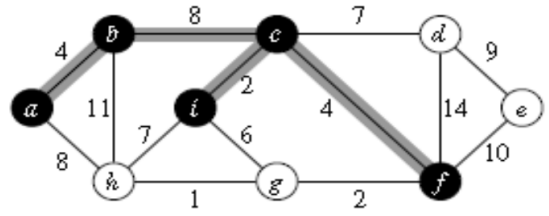
	a	b	c	d	e	f	g	h	i
π	*	a	b	c	*	c	*	a	c
key	0	4	8	7	∞	4	∞	8	2
Q				1	1	1	1	1	1

(d)



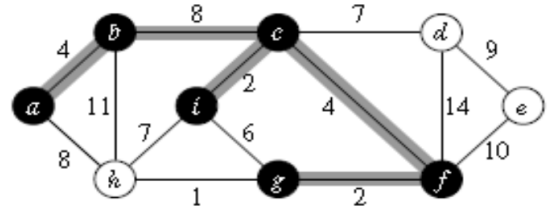
	a	b	c	d	e	f	g	h	i
π	*	a	b	c	*	c	i	i	c
key	0	4	8	7	∞	4	6	7	2
Q				1	1	1	1	1	

(e)



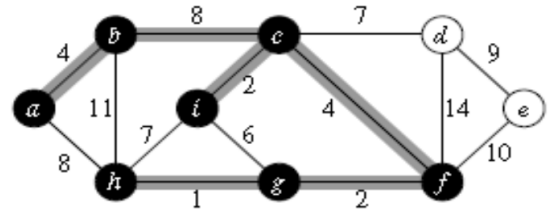
	a	b	c	d	e	f	g	h	i
π	*	a	b	c	f	c	f	i	c
key	0	4	8	7	10	4	2	7	2
Q				1	1		1	1	

(f)



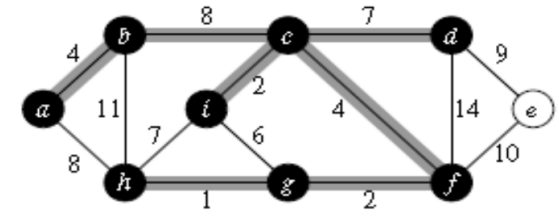
	a	b	c	d	e	f	g	h	i
π	*	a	b	c	f	c	f	g	c
key	0	4	8	7	10	4	2	1	2
Q				1	1			1	

(g)



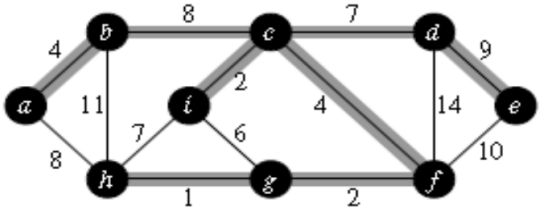
	a	b	c	d	e	f	g	h	i
π	*	a	b	c	f	c	f	g	c
key	0	4	8	7	10	4	2	1	2
Q				1	1				

(h)



	a	b	c	d	e	f	g	h	i
π	*	a	b	c	f	c	f	g	c
key	0	4	8	7	10	4	2	1	2
Q					1				

(i)



	a	b	c	d	e	f	g	h	i
π	*	a	b	c	f	c	f	g	c
key	0	4	8	7	10	4	2	1	2
Q									