Teoria dos Grafos

Berilhes

Árvores Geradoras Mínimas

Árvores Geradoras Mínimas

- Em 1920 o matemático Otakar Borůvka foi contratado para projetar uma rede elétrica eficiente para uma região do sul da república Checa.

Árvores Geradoras Mínimas

- Em 1920 o matemático Otakar Borůvka foi contratado para projetar uma rede elétrica eficiente para uma região do sul da república Checa.
- Borůvka modelou o problema como sendo um problema em grafos,

Árvores Geradoras Mínimas

- Em 1920 o matemático Otakar Borůvka foi contratado para projetar uma rede elétrica eficiente para uma região do sul da república Checa.
- Borůvka modelou o problema como sendo um problema em grafos,
- no qual cada vértice representa uma cidade que deve ser alimentada pela rede elétrica,

Árvores Geradoras Mínimas

- Em 1920 o matemático Otakar Borůvka foi contratado para projetar uma rede elétrica eficiente para uma região do sul da república Checa.
- Borůvka modelou o problema como sendo um problema em grafos,
- no qual cada vértice representa uma cidade que deve ser alimentada pela rede elétrica,
- e cada aresta $e = (u, v)$ possui um custo ou peso associado $c(e)$ que representa o custo de conectar as cidades u e v, por cabos de distribuição.

Árvore Geradora

Definição

Uma árvore geradora para um grafo conectado G é um subgrafo de G que é uma árvore contendo todo vértice de G. Se G não é conectado, um conjunto consistindo de uma árvore geradora para cada componente é chamado de floresta geradora.

Árvore Geradora

Definições

Árvore Geradora

Definições

(**Peso de uma árvore**) O peso de uma árvore é asoma dos pesos de suas arestas.

Árvore Geradora

Definições

(**Peso de uma árvore**) O peso de uma árvore é asoma dos pesos de suas arestas.
(**Árvore geradora mínima**) Uma árvore geradora mínima é uma árvore geradora de peso mínimo.

Árvore Geradora Mínima

Algoritmo Genérico

- Todos os algoritmos apresentados utilizam uma abordagem gulosa e

Árvore Geradora Mínima

Algoritmo Genérico

- Todos os algoritmos apresentados utilizam uma abordagem gulosa e
- podem ser entendidos como sendo instâncias algoritmo genérico.

Árvore Geradora Mínima

Algoritmo Genérico

- Todos os algoritmos apresentados utilizam uma abordagem gulosa e
- podem ser entendidos como sendo instâncias algoritmo genérico.

Árvore Geradora Mínima

Algoritmo Genérico

- Todos os algoritmos apresentados utilizam uma abordagem gulosa e
- podem ser entendidos como sendo instâncias algoritmo genérico.

MST-GENÉRICO(G)

1 $T \leftarrow \emptyset$
2 $Candidatas \leftarrow E$
3 **while** $(|T| \neq |V| - 1)$
4 Escolha uma aresta segura $e \in Candidatas$
5 $T \leftarrow T \cup \{e\}$
6 $Candidatas \leftarrow Candidatas - \{e\}$

Árvore Geradora Mínima

Algoritmo Genérico

- Como nós encontramos uma aresta segura?

Árvore Geradora Mínima

Algoritmo Genérico

- Como nós encontramos uma aresta segura?
- Cada vez que nós adicionamos uma aresta nõs estamos conectando dois conjuntos de vértices que anteriormente não estavam conectados.

Árvore Geradora Mínima

Algoritmo Genérico

- Como nós encontramos uma aresta segura?
- Cada vez que nós adicionamos uma aresta nõs estamos conectando dois conjuntos de vértices que anteriormente não estavam conectados.
- Caso contrário nós formaríamos um ciclo.

Árvore Geradora Mínima

Algoritmo Genérico

- Como nós encontramos uma aresta segura?
- Cada vez que nós adicionamos uma aresta nõs estamos conectando dois conjuntos de vértices que anteriormente não estavam conectados.
- Caso contrário nós formaríamos um ciclo.
- Um algoritmo guloso pode tentar manter os custos pequenos escolhendo a aresta de menor custo que conecta dois vértices não conectados.

Árvore Geradora Mínima

Definições

Árvore Geradora Mínima

Definições

(Conjunto Corte) S é um conjunto corte de um grafo conectado G se S é um conjunto minimal de arestas de G tal que $G - S$ é desconectado.

Árvore Geradora Mínima

Definições

(Conjunto Corte) S é um conjunto corte de um grafo conectado G se S é um conjunto minimal de arestas de G tal que $G - S$ é desconectado.

(Corte) V_1 e V_2 dois conjuntos distintos de V tal que $V_1 \cup V_2 = V$. Então o conjunto de arestas tendo uma extremidade em V_1 e outra em V_2 é chamado de corte de G.

Árvore Geradora Mínima

Definições

(Conjunto Corte) S é um conjunto corte de um grafo conectado G se S é um conjunto minimal de arestas de G tal que $G - S$ é desconectado.

(Corte) V_1 e V_2 dois conjuntos distintos de V tal que $V_1 \cup V_2 = V$. Então o conjunto de arestas tendo uma extremidade em V_1 e outra em V_2 é chamado de corte de G.

(Respeita) Um corte respeita um conjunto de arestas T sss nenhuma aresta de T cruza o corte.

Árvore Geradora Mínima

Definições

(Conjunto Corte) S é um conjunto corte de um grafo conectado G se S é um conjunto minimal de arestas de G tal que $G - S$ é desconectado.
(Corte) V_1 e V_2 dois conjuntos distintos de V tal que $V_1 \cup V_2 = V$. Então o conjunto de arestas tendo uma extremidade em V_1 e outra em V_2 é chamado de corte de G.
(Respeita) Um corte respeita um conjunto de arestas T sss nenhuma aresta de T cruza o corte.
(Aresta leve) Uma aresta (u, v) é uma aresta leve cruzando um corte sss seu peso é mínimo dentre todas as arestas cruzando o corte.

Árvore Geradora Mínima

Teorema

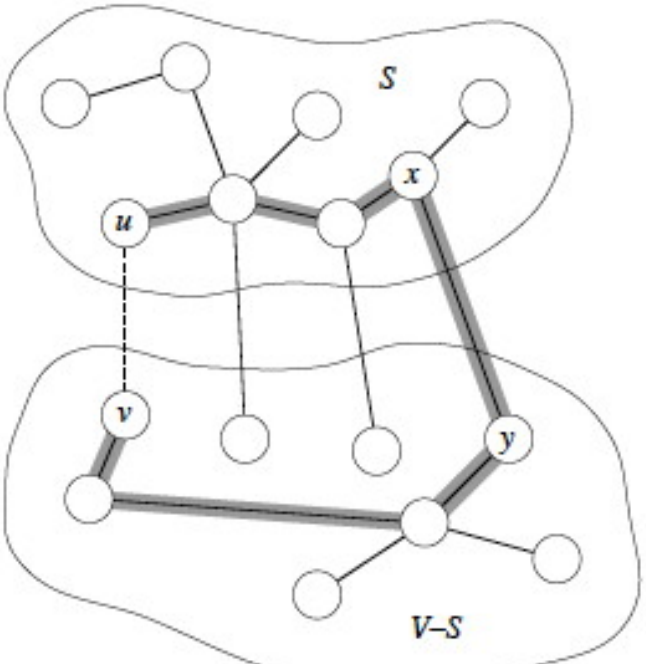
Assuma que $G = (V, E)$ *é um grafo,* T *é um subconjunto de alguma MST para* G*, e* $(S, V - S)$ *é um corte que respeita* T*, e* (u, v) *é uma aresta leve cruzando o corte* $(V, V - S)$*. Então* (u, v) *é uma aresta segura para* T*.*

Árvore Geradora Mínima

Teorema

Assuma que $G = (V, E)$ é um grafo, T é um subconjunto de alguma MST para G, e $(S, V - S)$ é um corte que respeita T, e (u, v) é uma aresta leve cruzando o corte $(V, V - S)$. Então (u, v) é uma aresta segura para T.

Demonstração.



Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

A Ideia

O algoritmo de Borůvka baseia-se na seguinte observação extremamente simples:

A árvore geradora mínima para G deve conter a aresta de peso mínima (v, w) que é a aresta de peso mínimo incidente no vértice v.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

A Ideia

O algoritmo de Borůvka baseia-se na seguinte observação extremamente simples:

A árvore geradora mínima para G deve conter a aresta de peso mínima (v, w) que é a aresta de peso mínimo incidente no vértice v.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- Contração é o processo de colapsar os dois pontos extremos de uma aresta em um único vértice

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- Contração é o processo de colapsar os dois pontos extremos de uma aresta em um único vértice
- que tem todas as arestas incidentes em ambos os vértices, auto-laços são eliminados.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- Contração é o processo de colapsar os dois pontos extremos de uma aresta em um único vértice
- que tem todas as arestas incidentes em ambos os vértices, auto-laços são eliminados.
- Contração pode criar arestas múltiplas entre alguns vértices mas somente a aresta de peso mínimo necessita ser mantida.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- Contração é o processo de colapsar os dois pontos extremos de uma aresta em um único vértice
- que tem todas as arestas incidentes em ambos os vértices, auto-laços são eliminados.
- Contração pode criar arestas múltiplas entre alguns vértices mas somente a aresta de peso mínimo necessita ser mantida.
- A ideia do algoritmo de Borůvka é contrair simultaneamente as arestas de peso mínimo incidentes sobre cada vértice em G.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- Contração é o processo de colapsar os dois pontos extremos de uma aresta em um único vértice
- que tem todas as arestas incidentes em ambos os vértices, auto-laços são eliminados.
- Contração pode criar arestas múltiplas entre alguns vértices mas somente a aresta de peso mínimo necessita ser mantida.
- A ideia do algoritmo de Borůvka é contrair simultaneamente as arestas de peso mínimo incidentes sobre cada vértice em G.
- O processo de contrair a aresta peso mínimo incidente em cada vértice é chamado de *fase de* Borůvka.

Árvore Geradora Mínima

Algoritmo de Borůvkal (cont.)

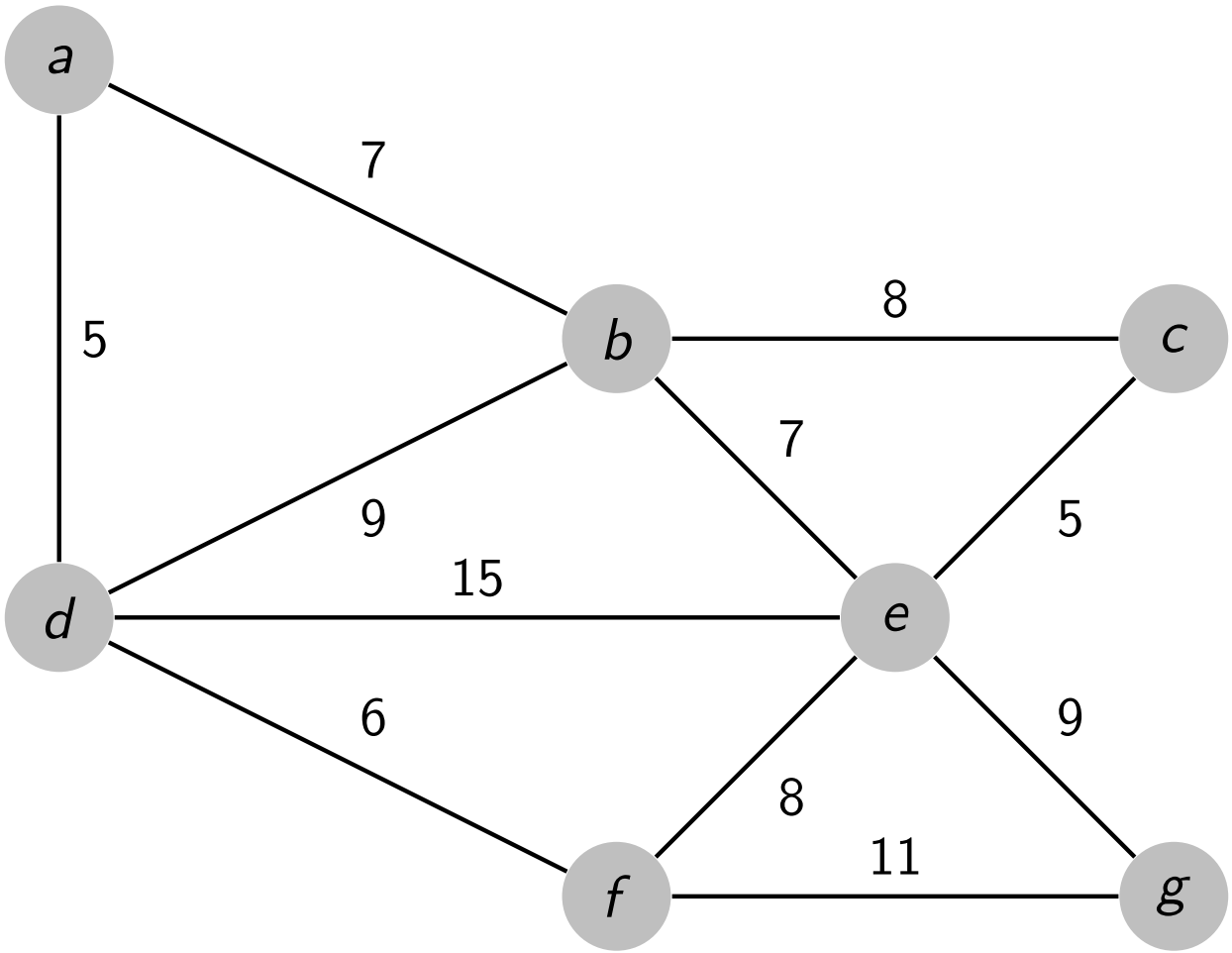
Árvore Geradora Mínima

Algoritmo de Borůvkal (cont.)

```
BORUVKA(G, w)
1  T ← ∅
   // Cada iteração do laço while é chamado de uma Fase
2  while G contém mais de um componente
3        Marque as arestas com peso mínimo incidentes
         em cada vértice. Adicione estas arestas à T
4        Identifique os componentes conectados das arestas
         marcadas
5        Substitua cada componente conectado por um vértice
6        Elimine todos os auto-laços.
7        Elimine todas as arestas múltiplas entre os
         pares de vértices, exceto a aresta de menor peso.
8  return T
```

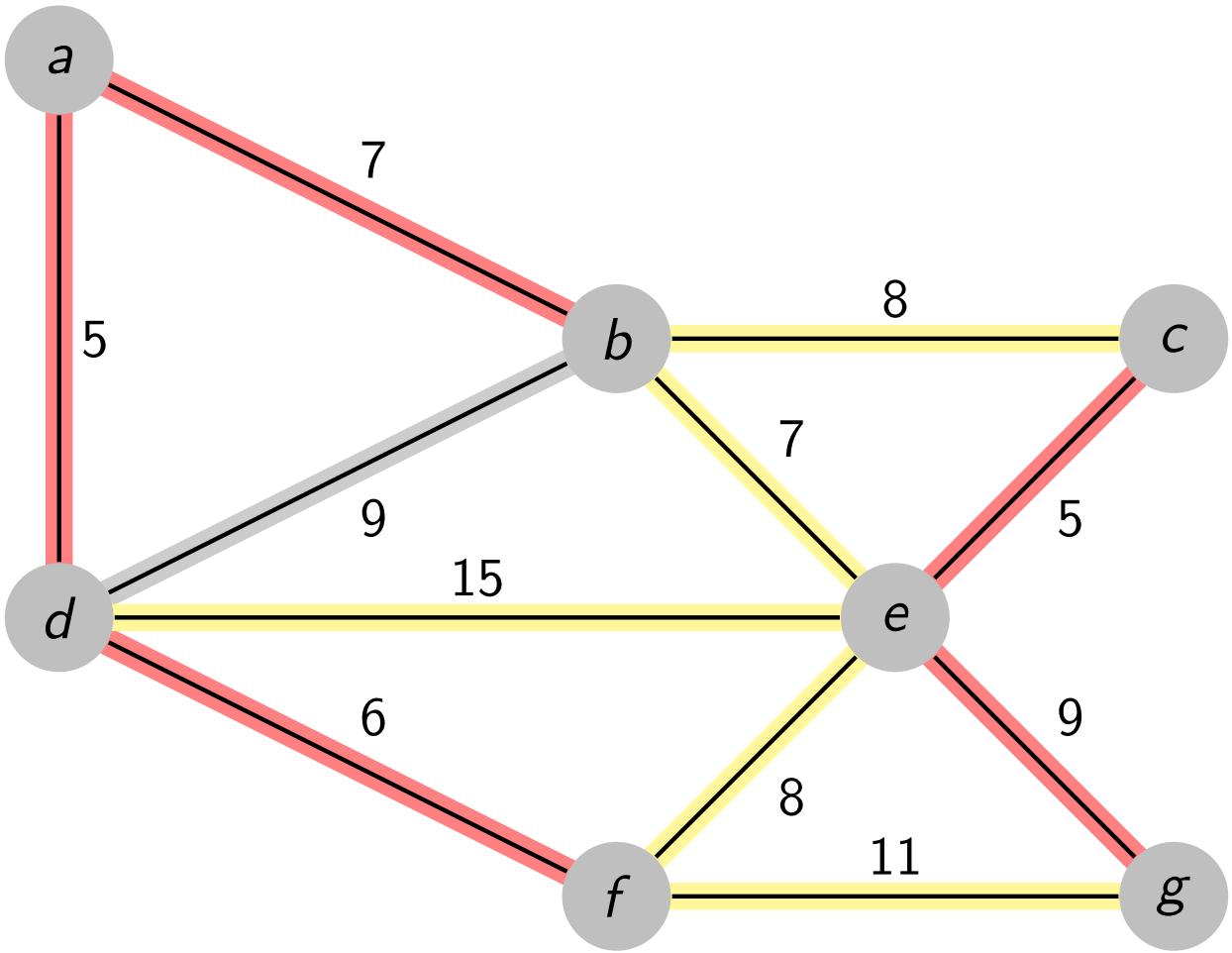
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



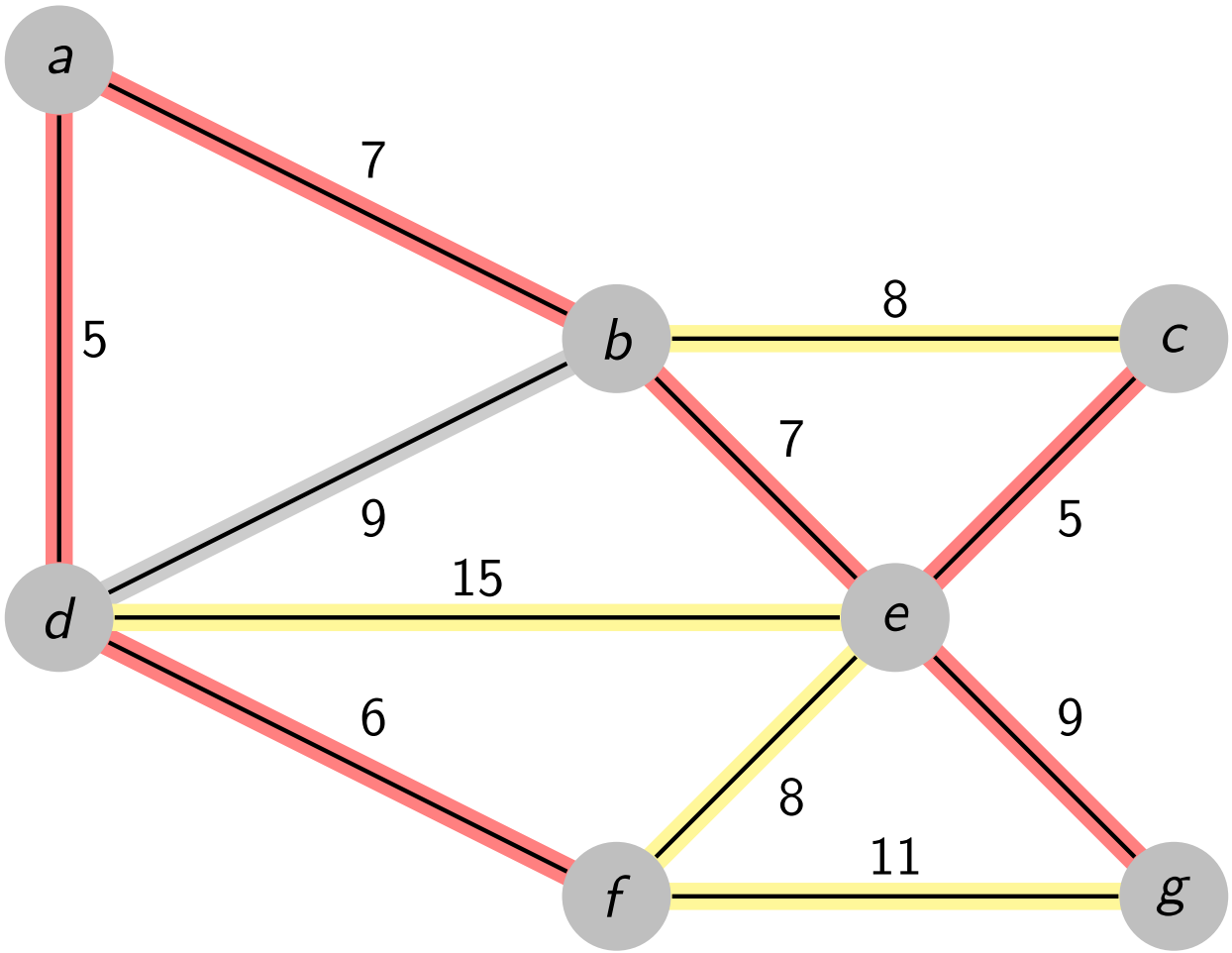
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



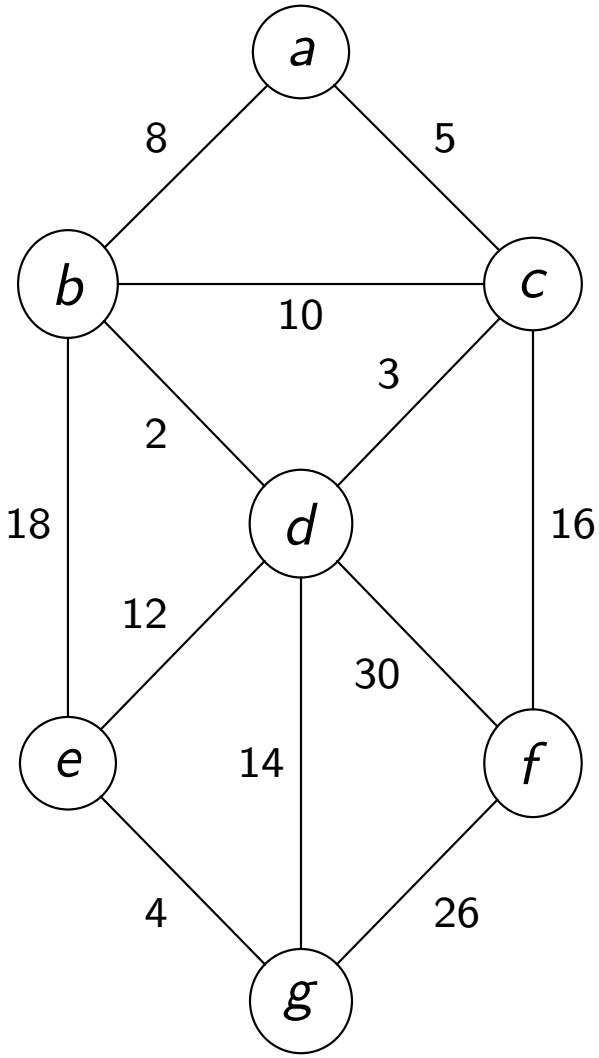
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



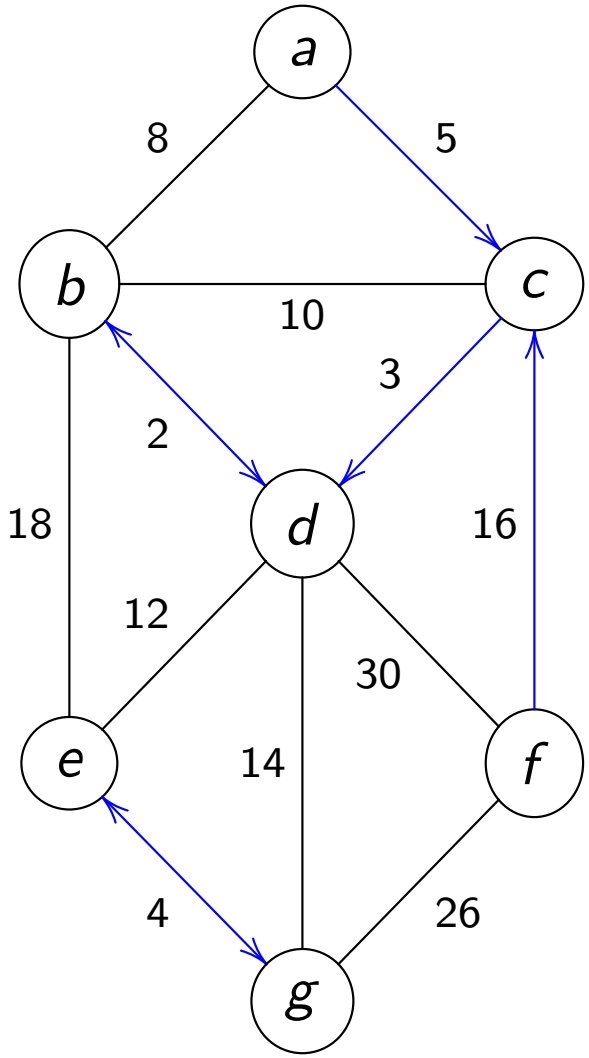
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



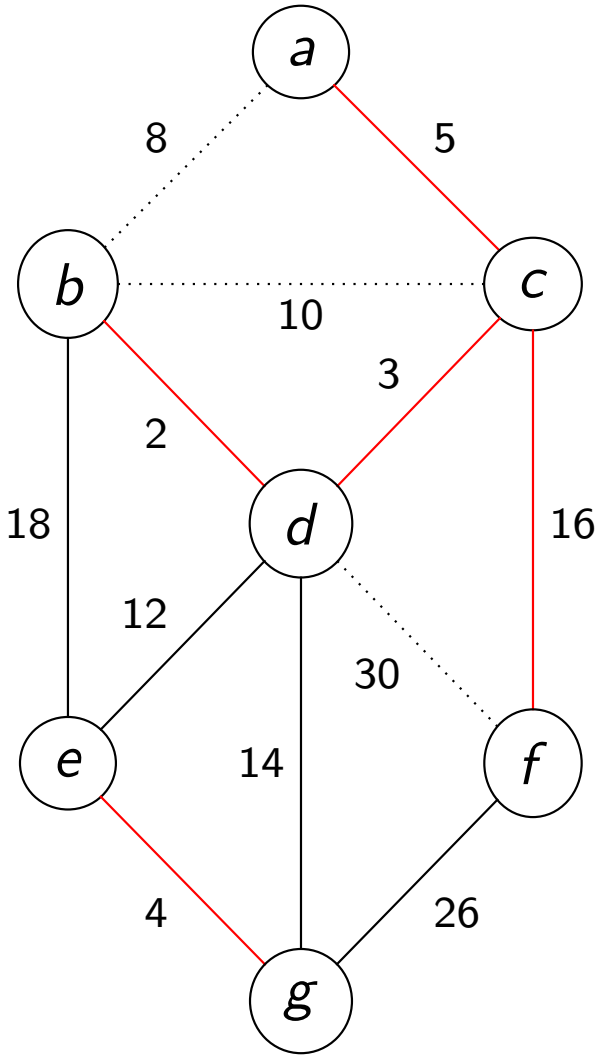
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



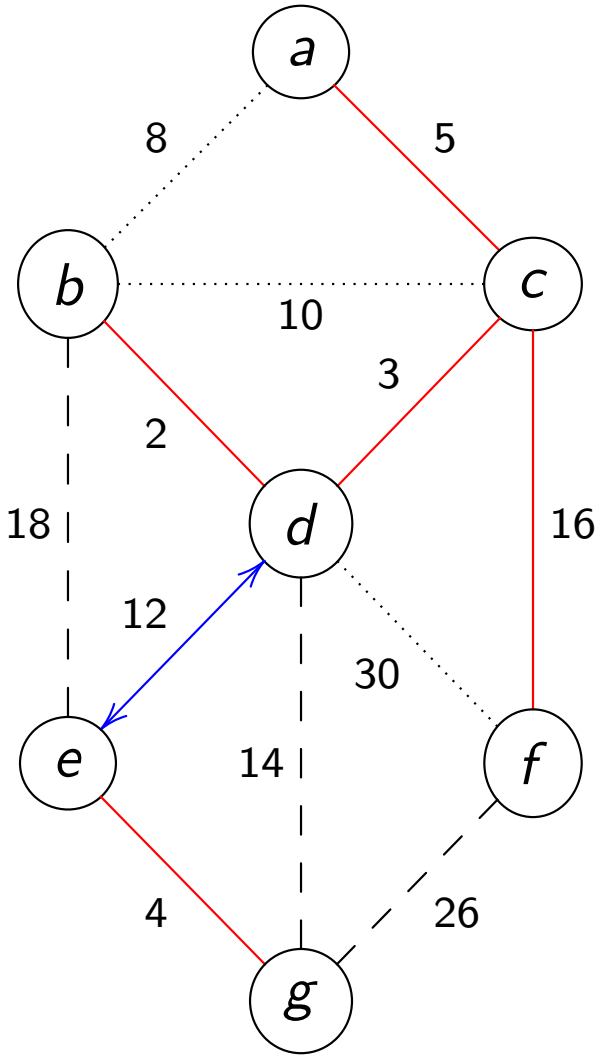
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



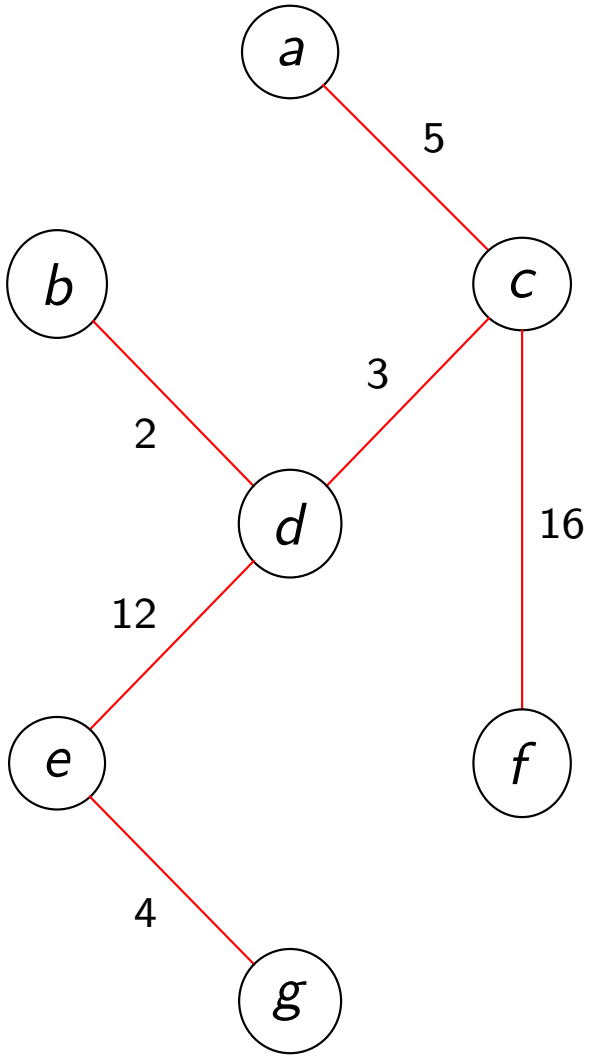
Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)



Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- O grafo obtido G' obtido da fase de Borůvka tem no máximo $n/2$ vértices.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- O grafo obtido G' obtido da fase de Borůvka tem no máximo $n/2$ vértices.
- Assim o número de arestas marcadas para contração é pelo menos $n/2$.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- O grafo obtido G' obtido da fase de Borůvka tem no máximo $n/2$ vértices.
- Assim o número de arestas marcadas para contração é pelo menos $n/2$.
- Portanto existirão somente $O(\lg n)$ fases de Borůvka.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

- O grafo obtido G' obtido da fase de Borůvka tem no máximo $n/2$ vértices.
- Assim o número de arestas marcadas para contração é pelo menos $n/2$.
- Portanto existirão somente $O(\lg n)$ fases de Borůvka.
- Cada fase pode ser implementada em tempo $O(m + n)$. Assim o tempo total do algoritmo é $O(m \lg n)$.

Vantagens de Borůvka:

- O algoritmo frequentemente executa mais rápido que $O(m \log n)$ no pior caso.

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

Vantagens de Borůvka:

- O algoritmo frequentemente executa mais rápido que $O(m \log n)$ no pior caso.
- O algoritmo permite um grau significativo de paralelismo

Árvore Geradora Mínima

Algoritmo de Borůvka (cont.)

Vantagens de Borůvka:

- O algoritmo frequentemente executa mais rápido que $O(m \log n)$ no pior caso.
- O algoritmo permite um grau significativo de paralelismo
- Vários algoritmos recentes são generalizações do algoritmo de Borůvka.

Árvore Geradora Mínima

Algoritmo de Kruskal

A Ideia

O algoritmo de Kruskal inspeciona todas as arestas em ordem crescente de peso, se uma aresta é segura, ela é então adicionada à F.

A Ideia

O algoritmo de Kruskal inspeciona todas as arestas em ordem crescente de peso, se uma aresta é segura, ela é então adicionada à F.

- Uma aresta é segura se e somente se suas extremidades estão em componentes diferentes na floresta F

Árvore Geradora Mínima

Algoritmo de Kruskal (cont.)

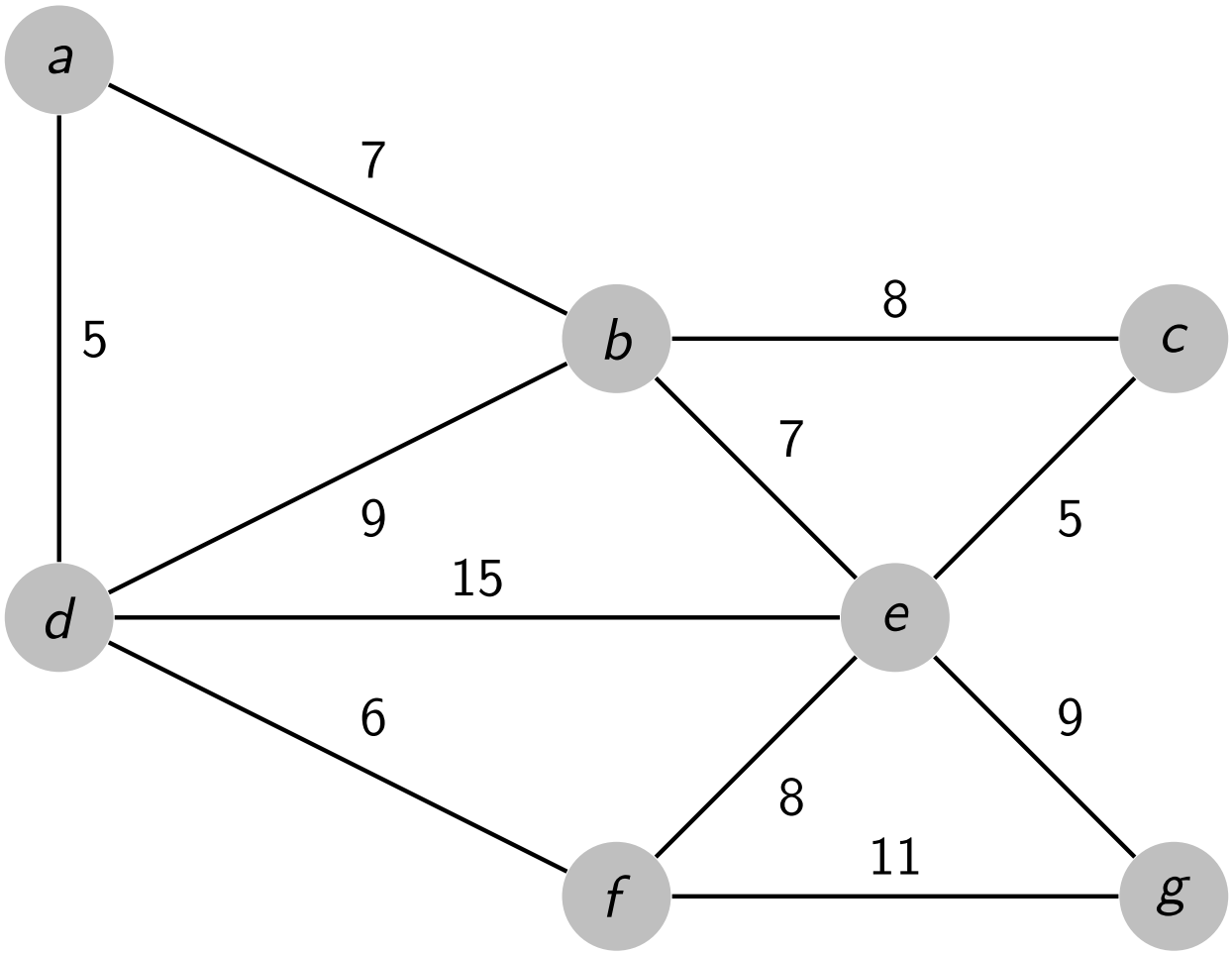
Árvore Geradora Mínima

Algoritmo de Kruskal (cont.)

```
KRUSKAL(V, E, w)
1  A ← ∅
2  for cada vértice v ∈ V
3        FAÇACONJUNTO(v)
4  ordene as arestas de E em ordem não decrescente de custo w
5  for cada aresta (u, v) tomada em ordem
6        if CONJUNTO(u) ≠ CONJUNTO(v)
7              UNIR(CONJUNTO(u), CONJUNTO(v))
8              A ← A ∪ {(u, v)}
9  return A
```

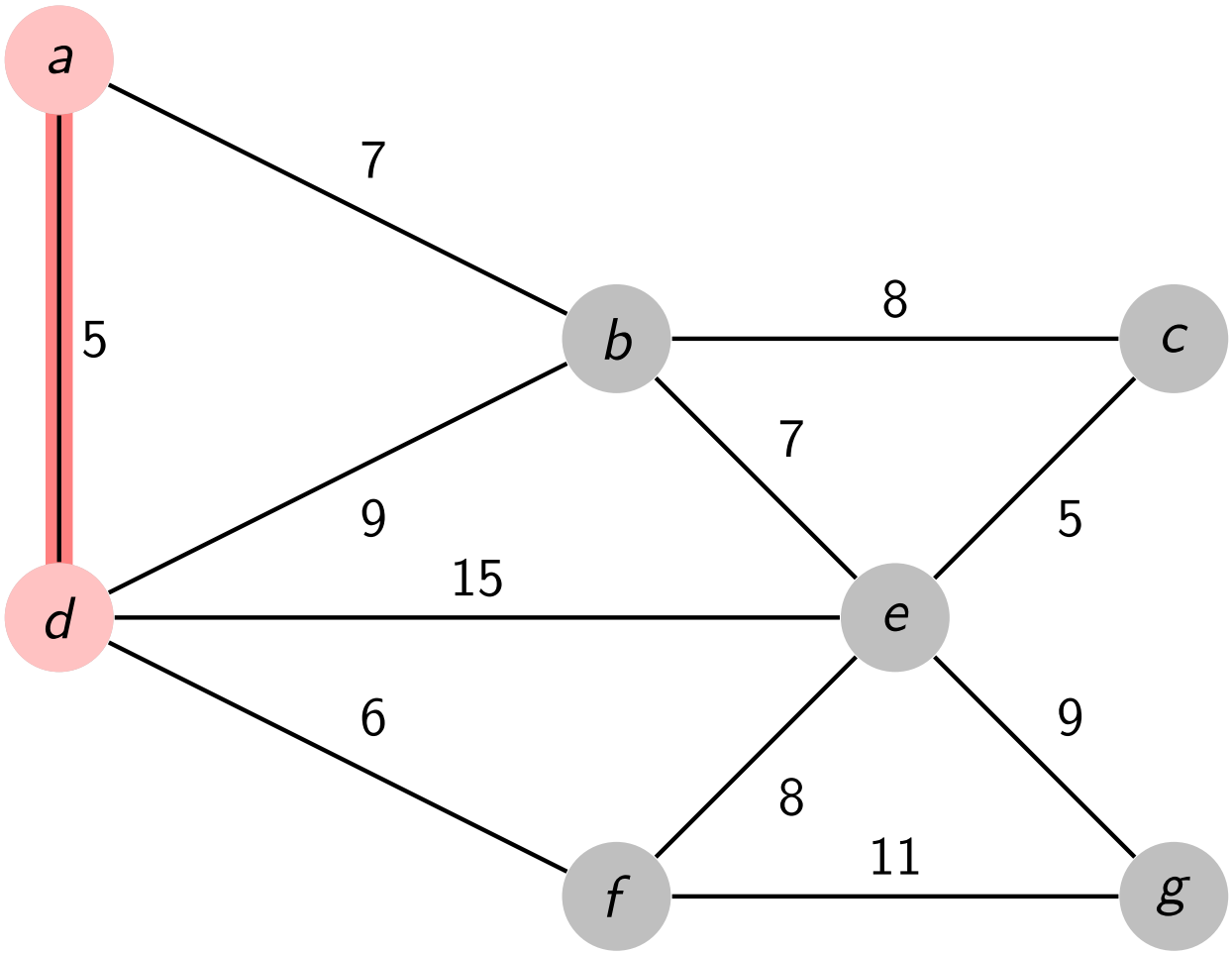
Árvore Geradora Mínima

Algoritmo de Kruskal



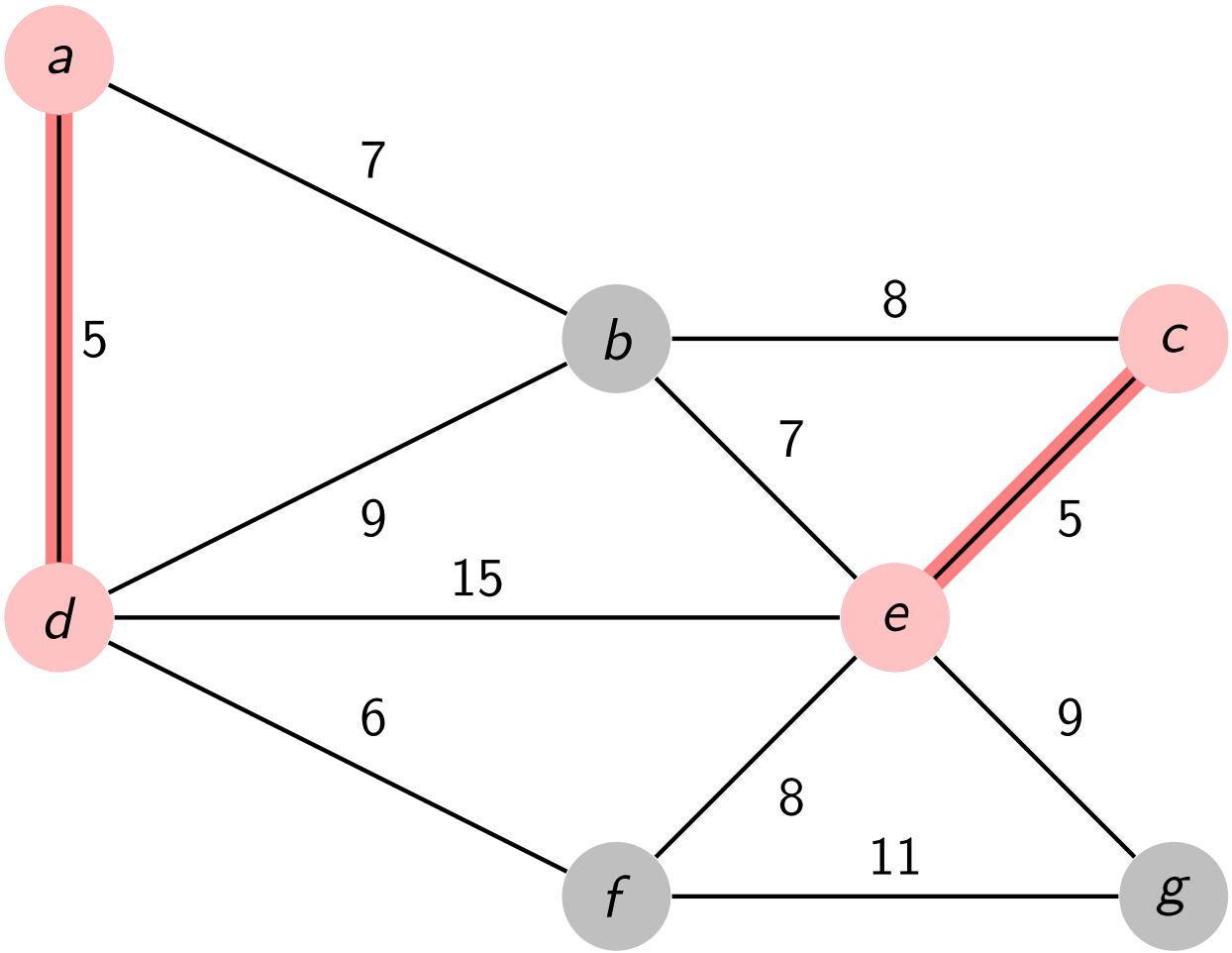
Árvore Geradora Mínima

Algoritmo de Kruskal



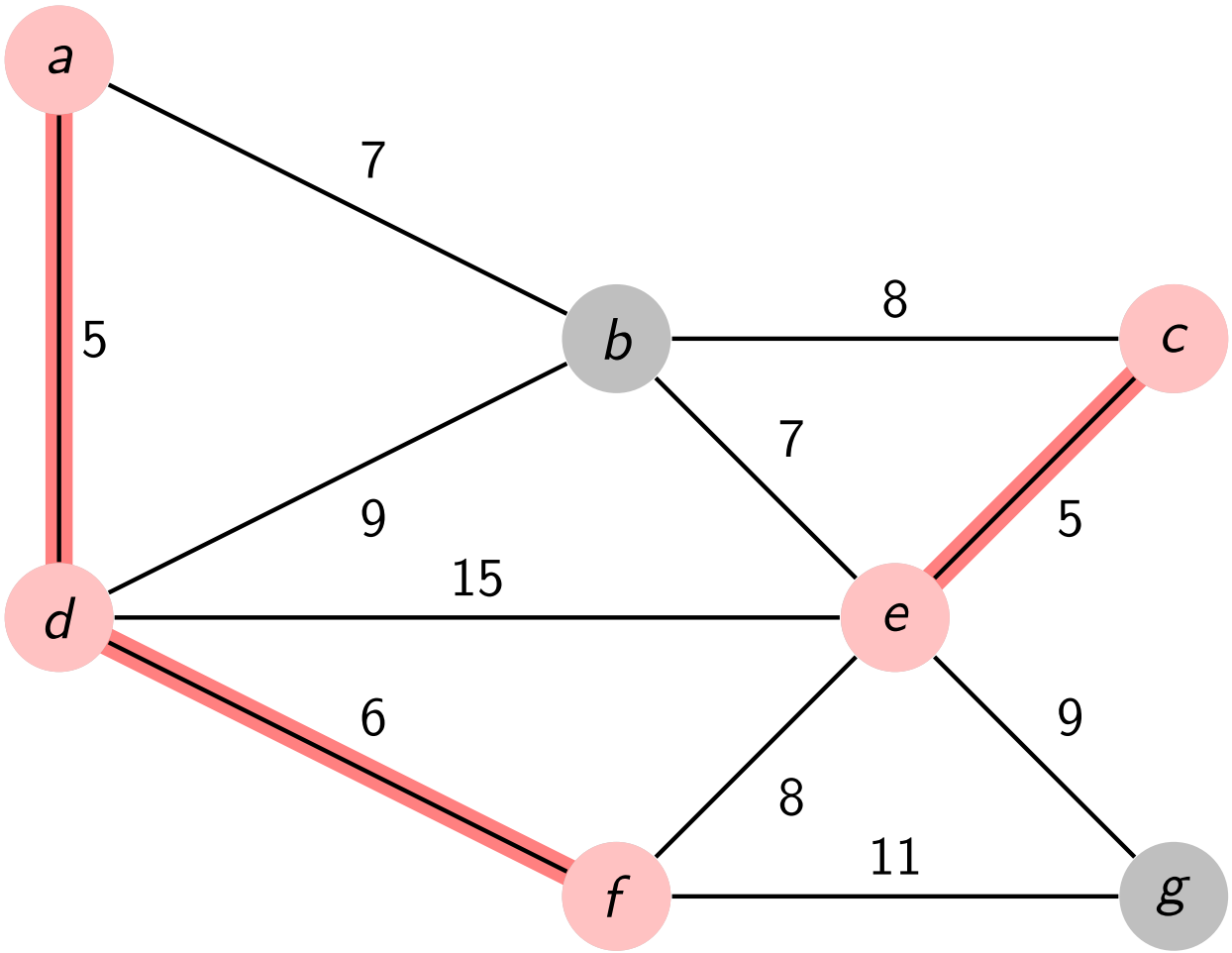
Árvore Geradora Mínima

Algoritmo de Kruskal



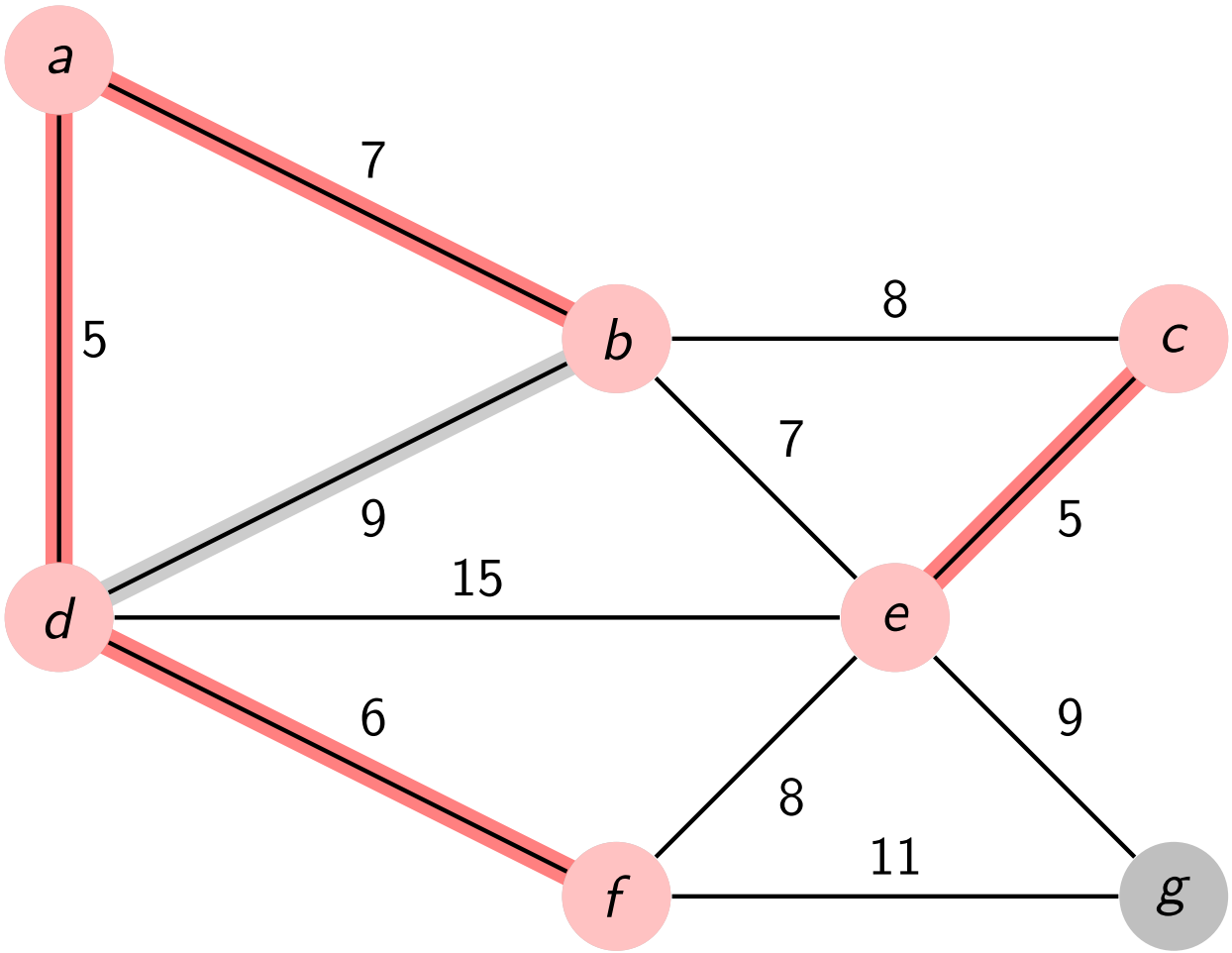
Árvore Geradora Mínima

Algoritmo de Kruskal



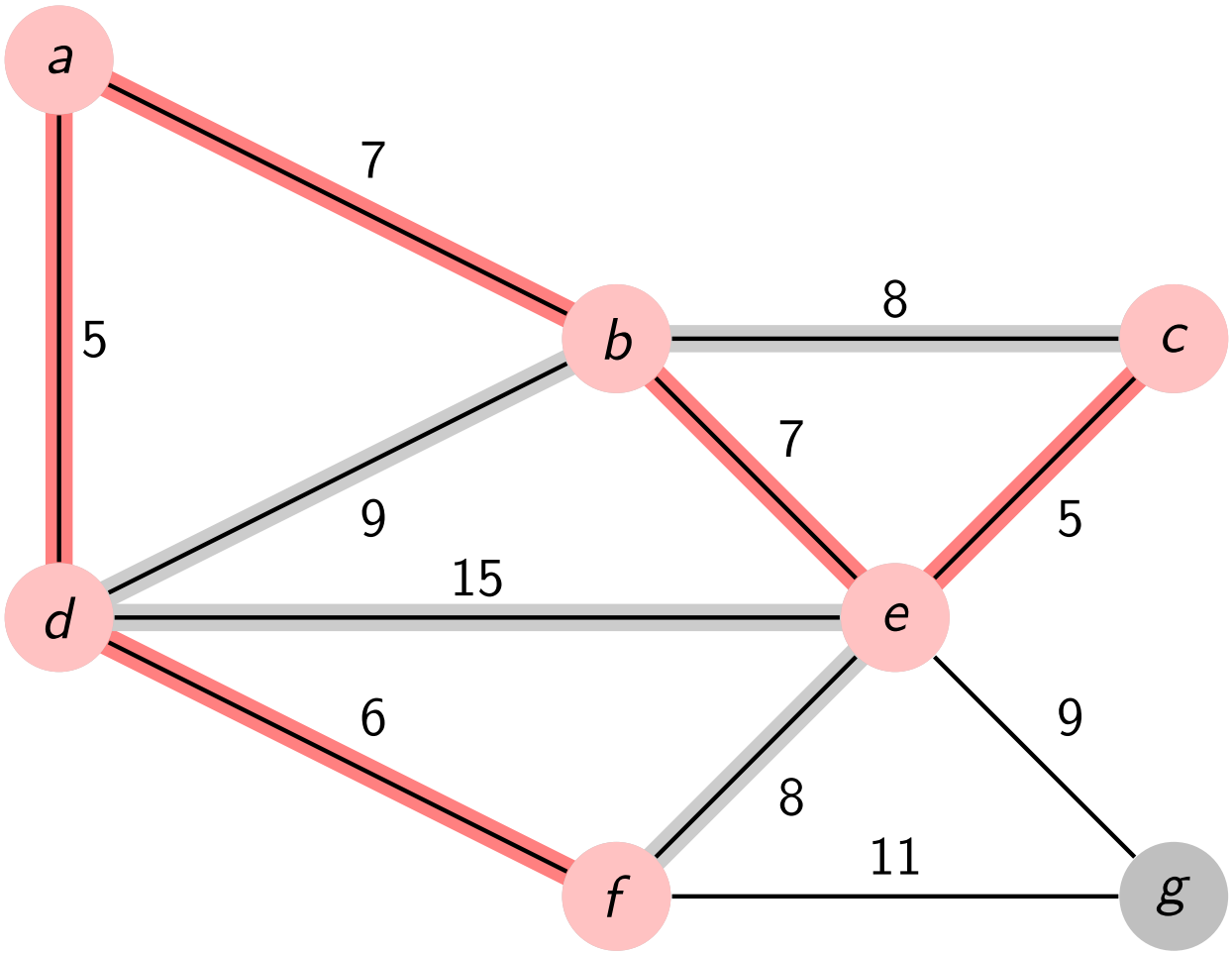
Árvore Geradora Mínima

Algoritmo de Kruskal



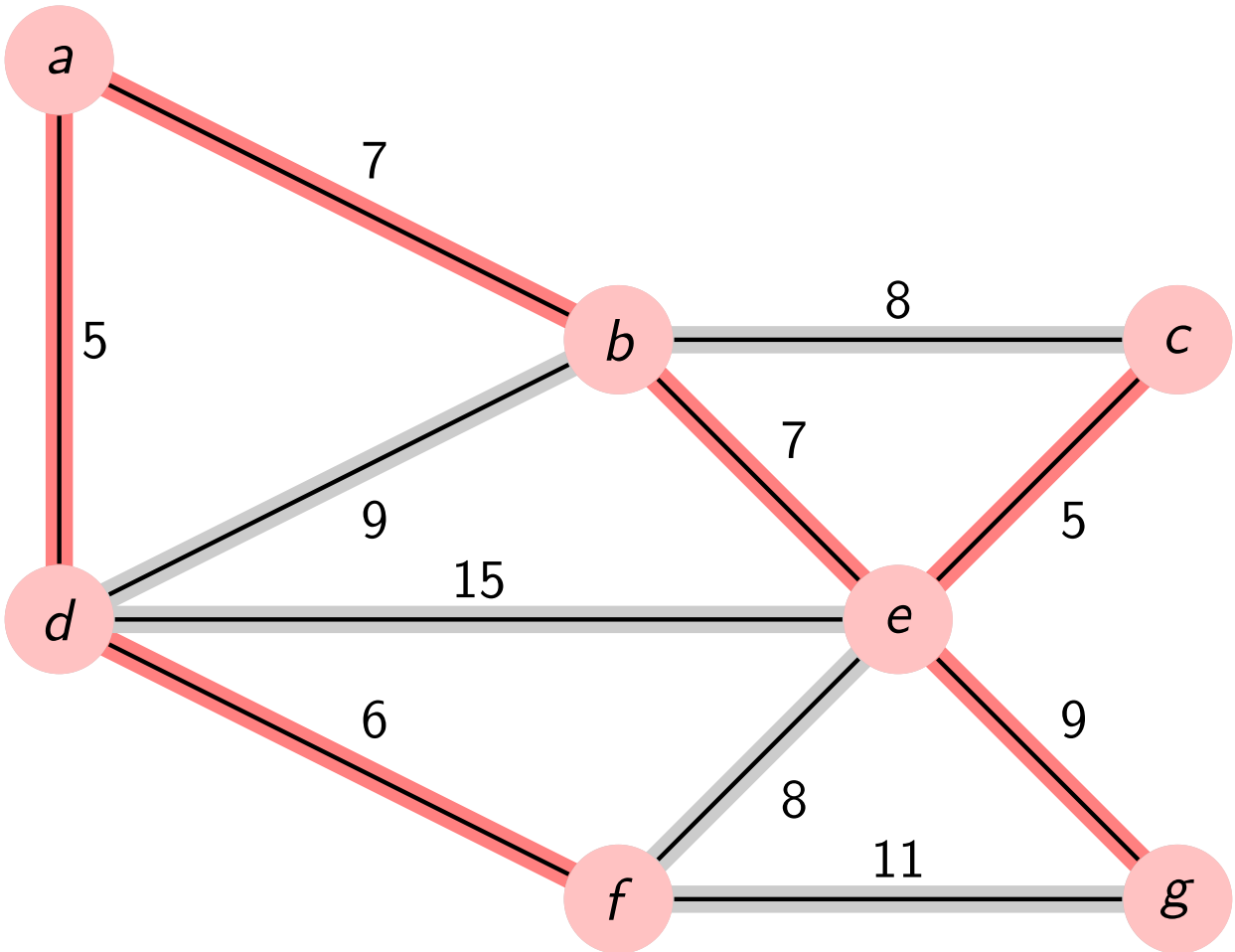
Árvore Geradora Mínima

Algoritmo de Kruskal



Árvore Geradora Mínima

Algoritmo de Kruskal



Árvore Geradora Mínima

Algoritmo de Kruskal

- Os conjuntos são os componentes de F.

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,
- e $O(n)$ operações Unir, uma para cada aresta na árvore geradora mínima.

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,
- e $O(n)$ operações Unir, uma para cada aresta na árvore geradora mínima.
- Pode-se executar cada operação Encontrar e Unir em tempo $O(\alpha(m, n))$.

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,
- e $O(n)$ operações Unir, uma para cada aresta na árvore geradora mínima.
- Pode-se executar cada operação Encontrar e Unir em tempo $O(\alpha(m, n))$.
- Assim ignorando o custo de ordenar as arestas, o tempo de execução do algoritmo de KRUSKAL é $O(m\alpha(m.n))$.

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,
- e $O(n)$ operações Unir, uma para cada aresta na árvore geradora mínima.
- Pode-se executar cada operação Encontrar e Unir em tempo $O(\alpha(m, n))$.
- Assim ignorando o custo de ordenar as arestas, o tempo de execução do algoritmo de KRUSKAL é $O(m\alpha(m.n))$.
- O tempo para ordenar as arestas é $O(m \log m) = O(m \log n)$.

- Os conjuntos são os componentes de F.
- O algoritmo de KRUSKAL executa $O(m)$ operações ENCONTRAR, duas para cada aresta no grafo,
- e $O(n)$ operações Unir, uma para cada aresta na árvore geradora mínima.
- Pode-se executar cada operação Encontrar e Unir em tempo $O(\alpha(m, n))$.
- Assim ignorando o custo de ordenar as arestas, o tempo de execução do algoritmo de KRUSKAL é $O(m\alpha(m.n))$.
- O tempo para ordenar as arestas é $O(m \log m) = O(m \log n)$.
- O algoritmo de KRUSKAL executa em tempo $O(m \lg n)$.

Árvore Geradora Mínima

Algoritmo de Prim

A Ideia

Repetidamente adicione uma aresta segura à $\mathcal{T}$.

A Ideia

Repetidamente adicione uma aresta segura à $\mathcal{T}$.

- No algoritmo de Prim, a floresta F contem somente um componente $\mathcal{T}$, todos os outros componentes são vértices isolados.

A Ideia

Repetidamente adicione uma aresta segura à $\mathcal{T}$.

- No algoritmo de Prim, a floresta F contem somente um componente $\mathcal{T}$, todos os outros componentes são vértices isolados.
- No começo, $\mathcal{T}$ consiste de um vértice arbitrário do grafo.

Árvore Geradora Mínima

Algoritmo de Prim (cont.)

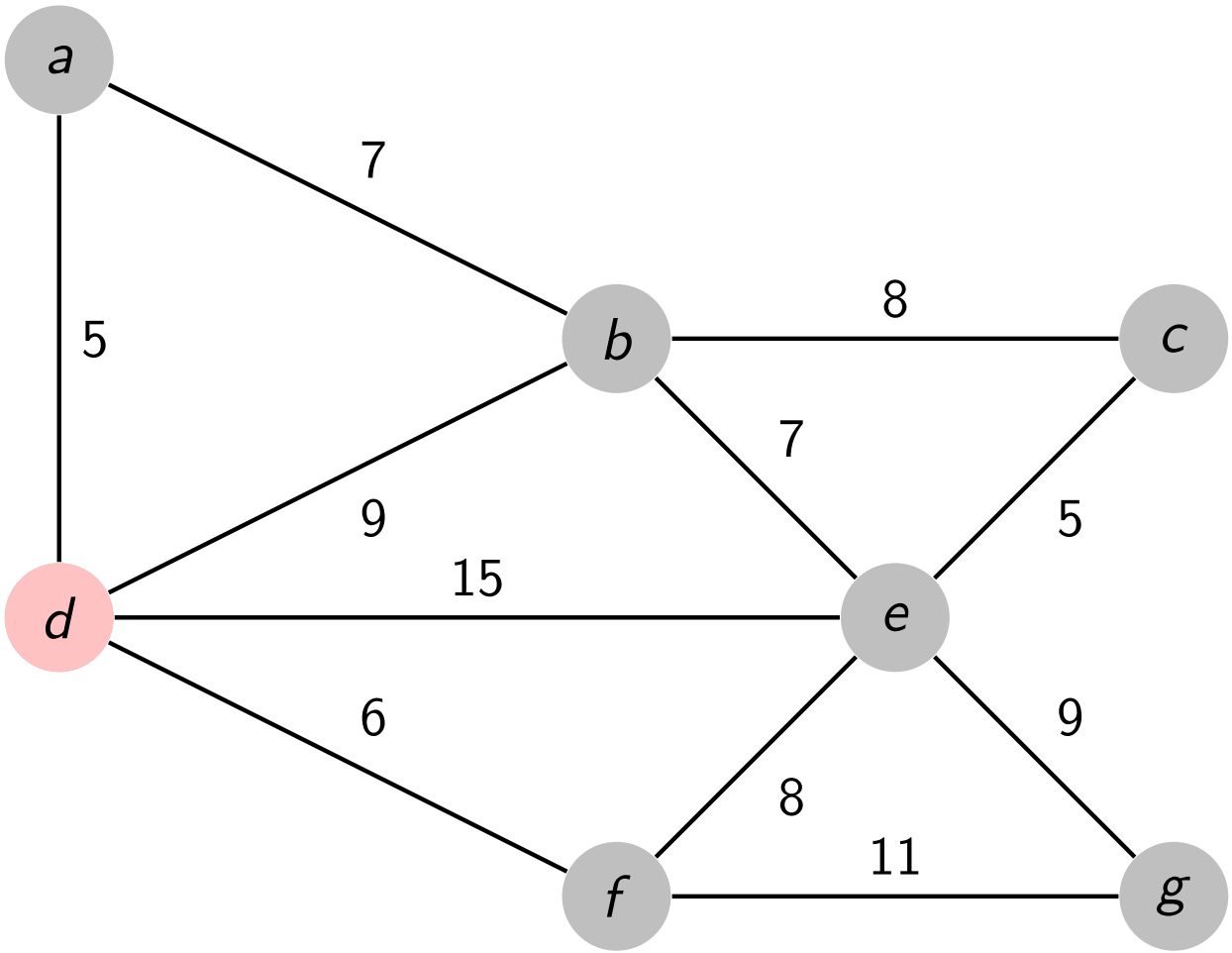
Árvore Geradora Mínima

Algoritmo de Prim (cont.)

```
PRIM(G, w, s)
 1  Q ← G.V
 2  for cada u ∈ Q
 3        u.chave ← ∞
 4  s.chave ← 0
 5  s.pai ← NIL
 6  while Q ≠ ∅
 7        u ← EXTRAIRMINIMO(Q)
 8        for cada vértice v ∈ u.Adj
 9              if v ∈ Q e w(u, v) < u.chave
10                    v.pai ← u
11                    v.chave ← w(u, v)
```

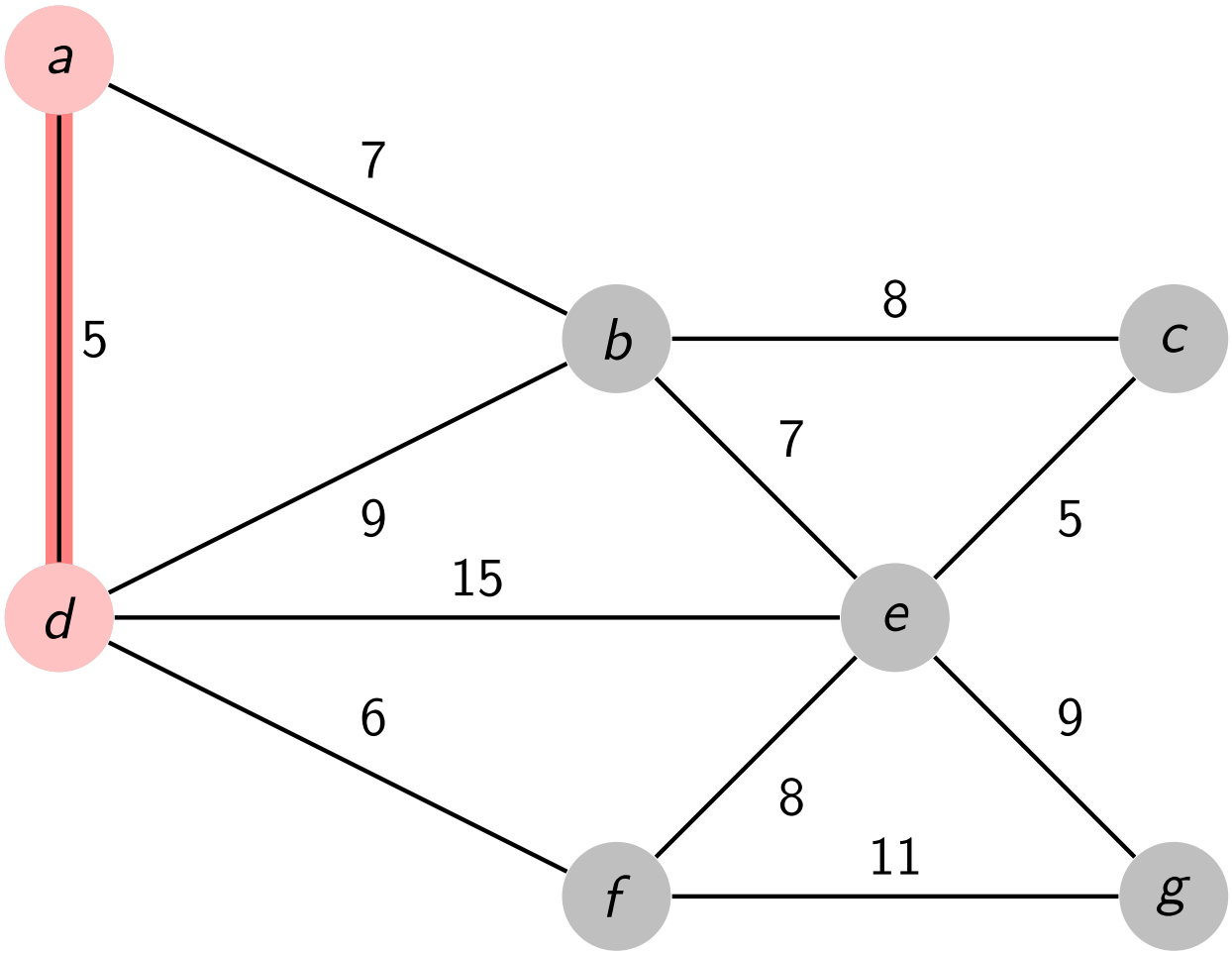
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



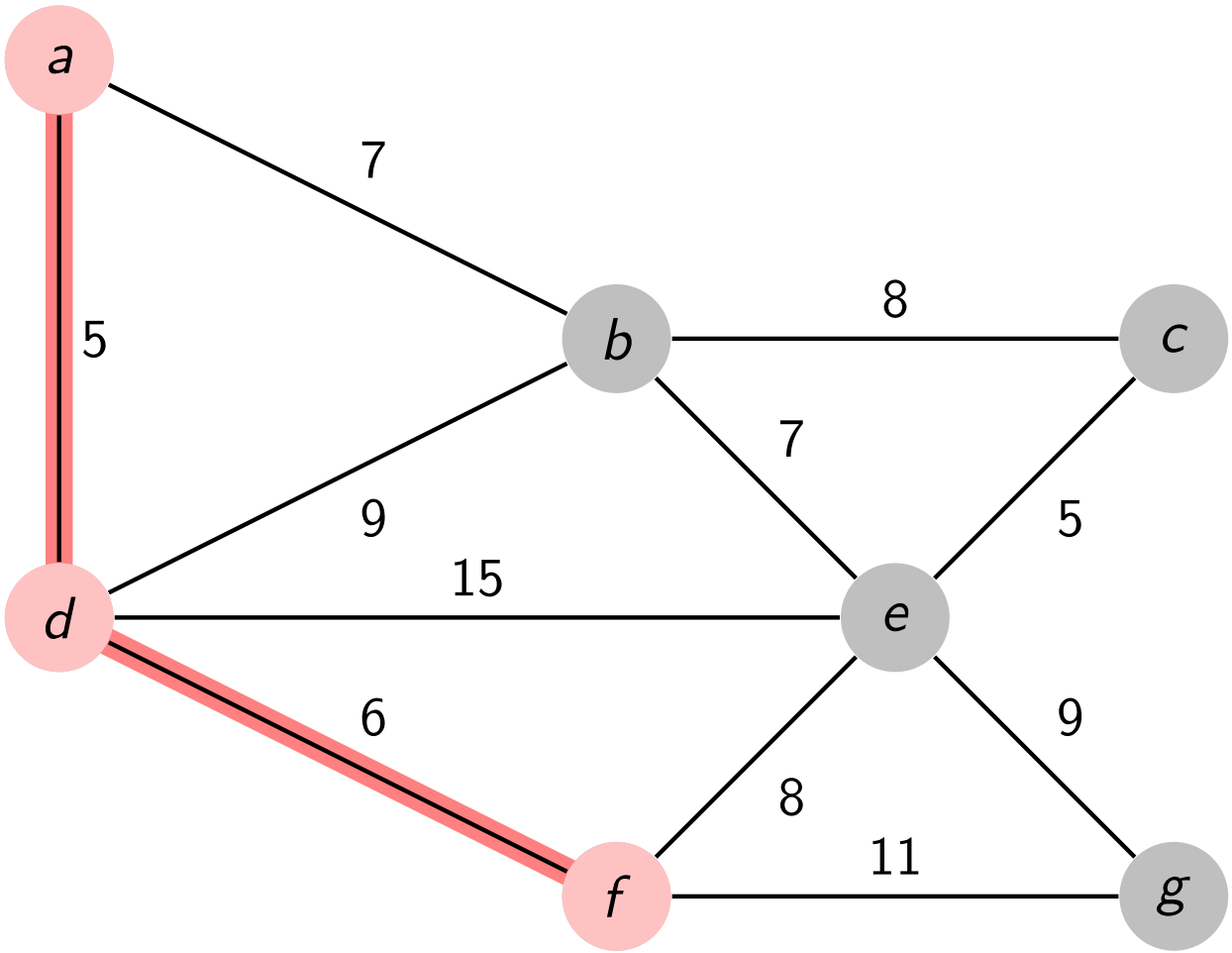
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



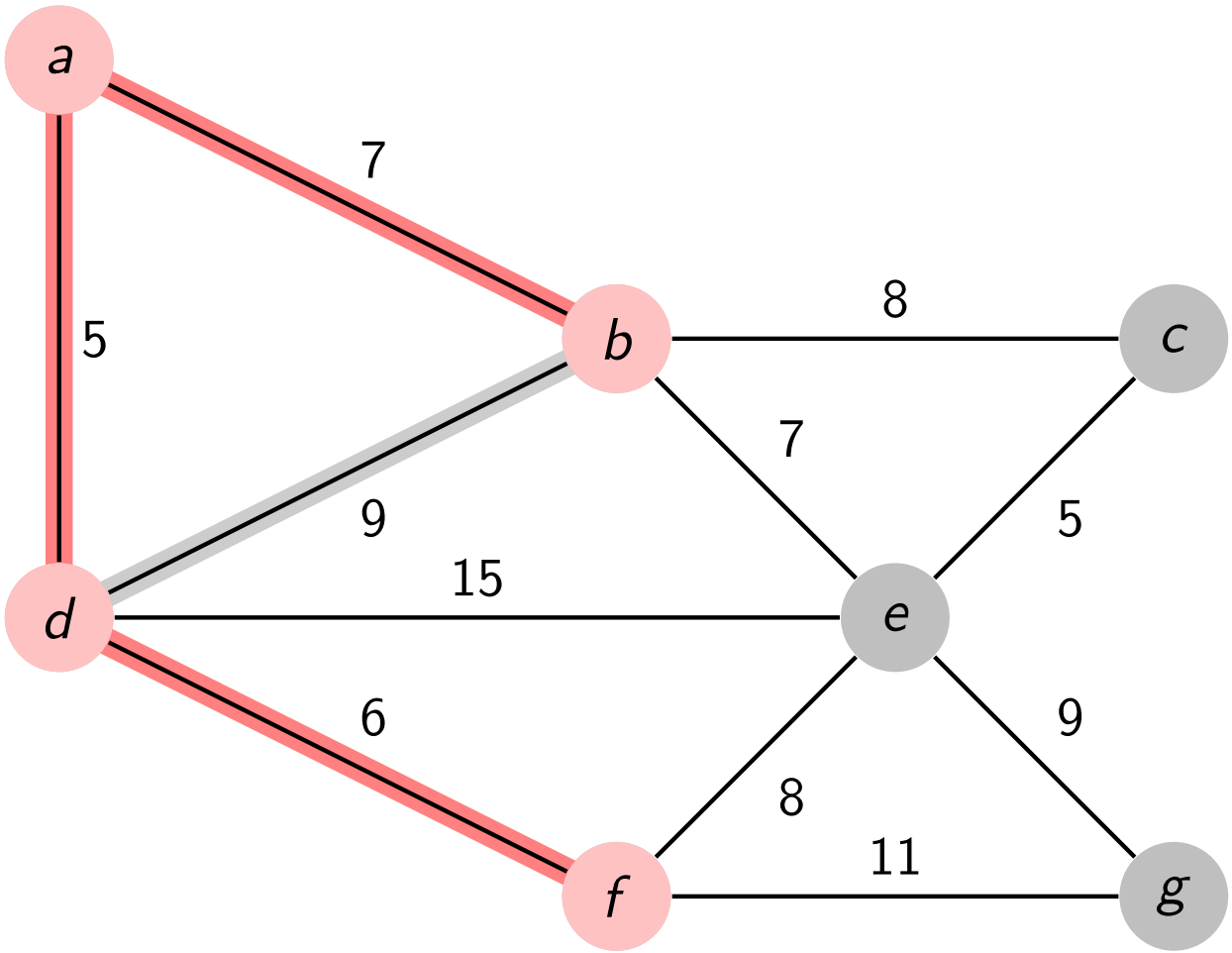
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



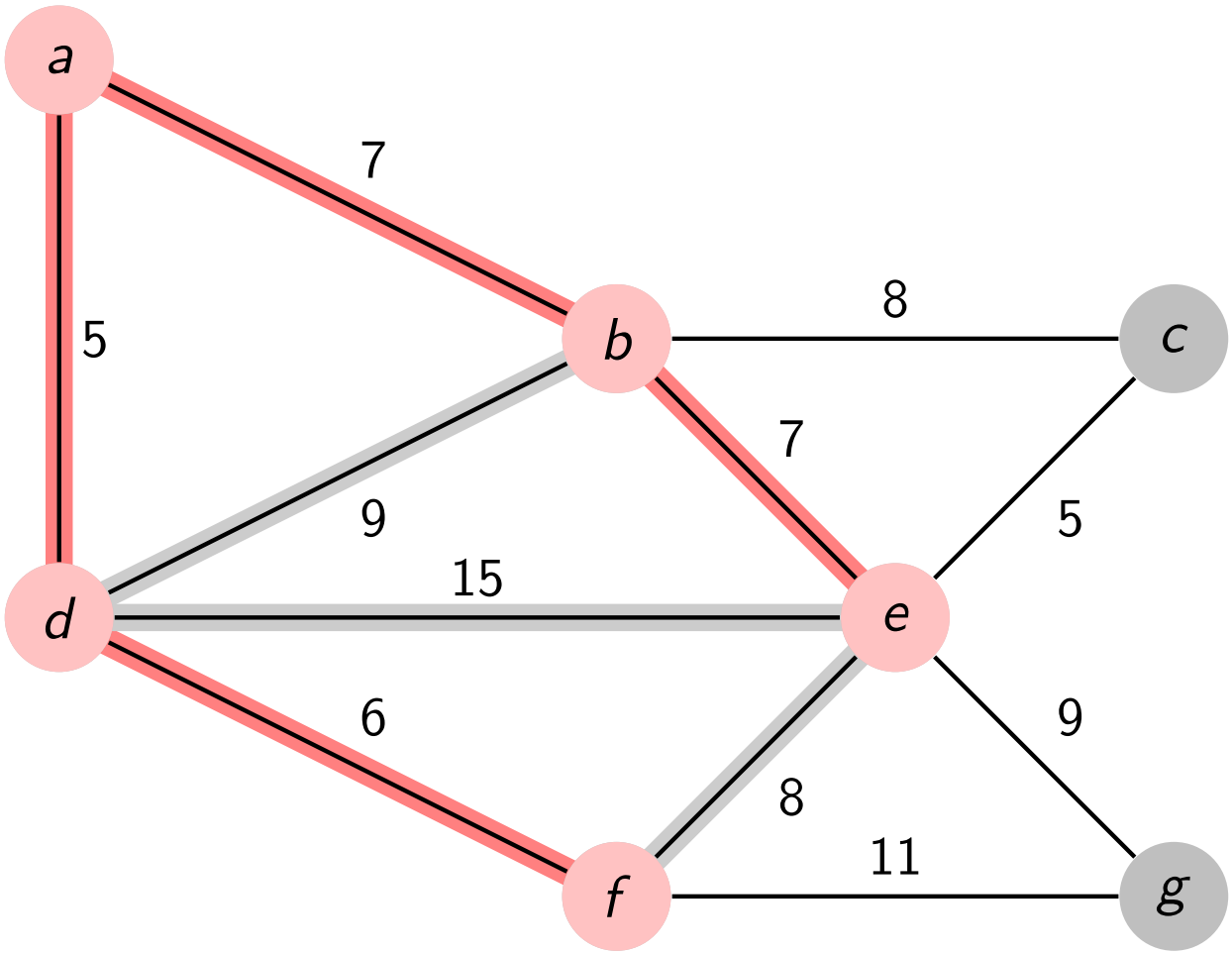
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



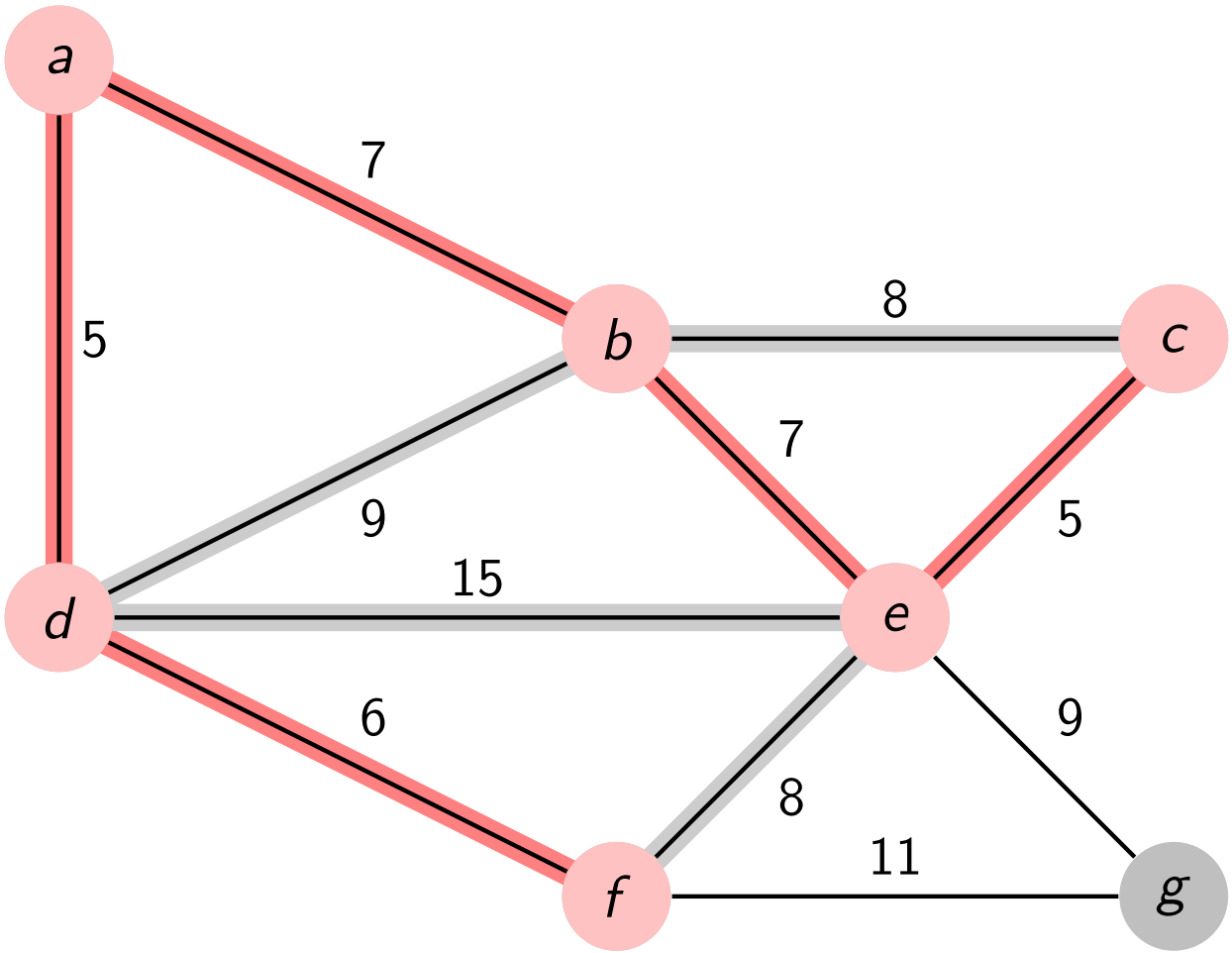
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



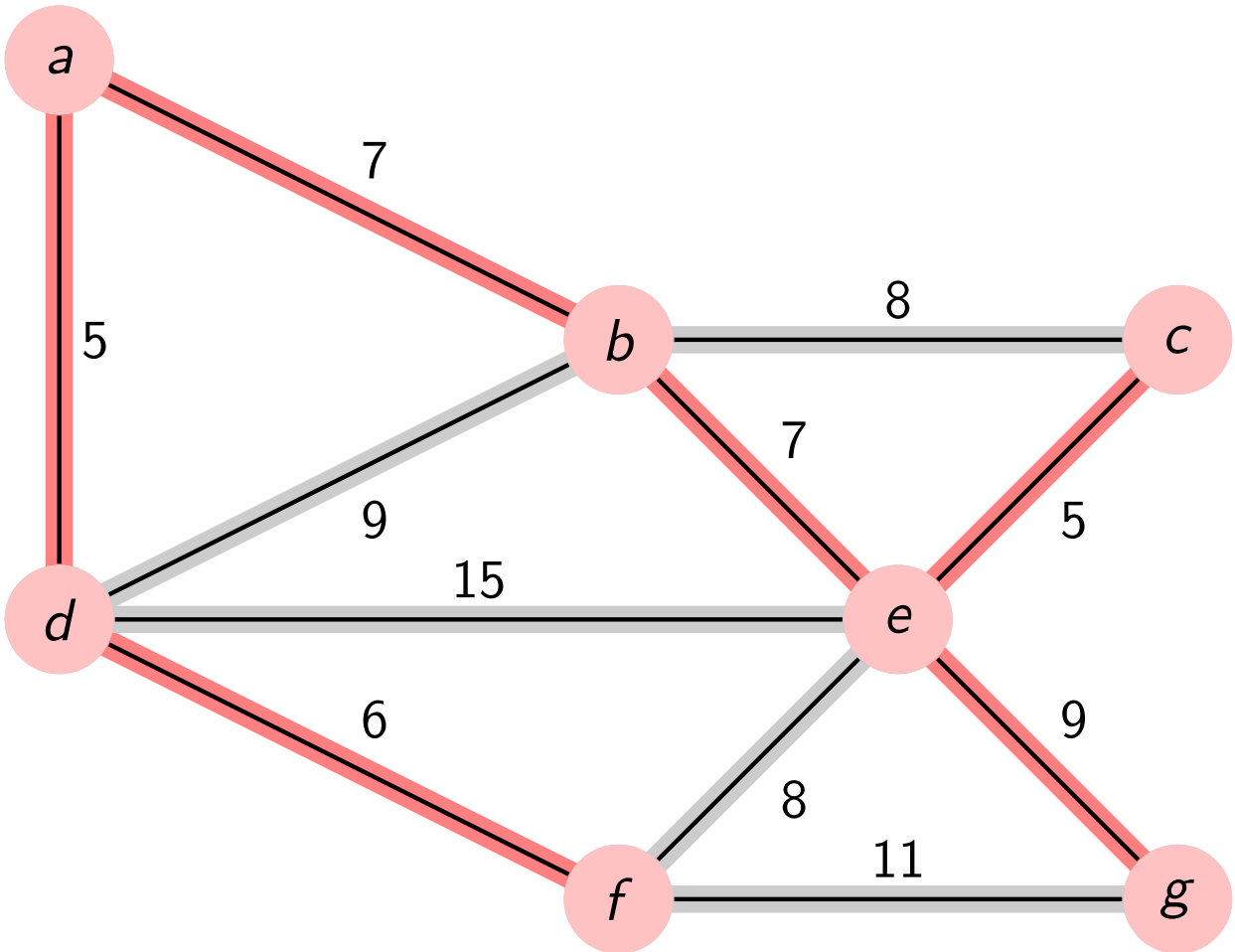
Árvore Geradora Mínima

Algoritmo de Prim (cont.)



Árvore Geradora Mínima

Algoritmo de Prim (cont.)



Árvore Geradora Mínima

Algoritmo de Prim (cont.)

Árvore Geradora Mínima

Algoritmo de Prim (cont.)

- Para implementar o algoritmo de Prim nós mantemos todas as arestas adjacentes à $\mathcal{T}$ em um fila com prioridade.

Árvore Geradora Mínima

Algoritmo de Prim (cont.)

- Para implementar o algoritmo de Prim nós mantemos todas as arestas adjacentes à $\mathcal{T}$ em um fila com prioridade.
- Quando nós retiramos a aresta de menor peso da fila com prioridade, nós primeiro verificamos se os dois pontos extremos da aresta estão em $\mathcal{T}$.

Árvore Geradora Mínima

Algoritmo de Prim (cont.)

- Para implementar o algoritmo de Prim nós mantemos todas as arestas adjacentes à $\mathcal{T}$ em um fila com prioridade.
- Quando nós retiramos a aresta de menor peso da fila com prioridade, nós primeiro verificamos se os dois pontos extremos da aresta estão em $\mathcal{T}$.
- Se não, nós adicionamos a aresta à $\mathcal{T}$ e então adicionamos as novas arestas adjacentes à $\mathcal{T}$ a fila com prioridade.

Árvore Geradora Mínima

Algoritmo de Prim (cont.)

- Para implementar o algoritmo de Prim nós mantemos todas as arestas adjacentes à $\mathcal{T}$ em um fila com prioridade.
- Quando nós retiramos a aresta de menor peso da fila com prioridade, nós primeiro verificamos se os dois pontos extremos da aresta estão em $\mathcal{T}$.
- Se não, nós adicionamos a aresta à $\mathcal{T}$ e então adicionamos as novas arestas adjacentes à $\mathcal{T}$ a fila com prioridade.
- Se nós implementamos o algoritmo dessa forma, o tempo de execução é $O(m \lg m) = O(m \lg n)$.