

Projeto e Análise de Algoritmos

Algoritmos Gulosos

Algoritmos Gulosos

- ▶ Imagine que você está com muita fome.

Algoritmos Gulosos

- ▶ Imagine que você está com muita fome.
- ▶ Na sua frente, há uma mesa com cookies.

Algoritmos Gulosos

- ▶ Imagine que você está com muita fome.
- ▶ Na sua frente, há uma mesa com cookies.
- ▶ Só que cada cookie tem um tamanho diferente.

Algoritmos Gulosos

- ▶ Imagine que você está com muita fome.
- ▶ Na sua frente, há uma mesa com cookies.
- ▶ Só que cada cookie tem um tamanho diferente.
- ▶ Por qual cookie você começa a comer?

Algoritmos Gulosos

- ▶ Algoritmos gulosos são “precipitados”.

Algoritmos Gulosos

- ▶ Algoritmos gulosos são “precipitados”.
- ▶ As escolhas são feitas localmente.

Algoritmos Gulosos

- ▶ Algoritmos gulosos são “precipitados”.
- ▶ As escolhas são feitas localmente.
- ▶ Escolhe-se a melhor solução no momento.

Algoritmos Gulosos

- ▶ Algoritmos gulosos são “precipitados”.
- ▶ As escolhas são feitas localmente.
- ▶ Escolhe-se a melhor solução no momento.
- ▶ E espera-se que as melhores escolhas locais resultem na melhor solução global.

Algoritmos Gulosos

- ▶ Lembra do problema do troco?

Algoritmos Gulosos

- ▶ Lembra do problema do troco?
- ▶ Um algoritmo guloso sempre chegava ao resultado.

Algoritmos Gulosos

- ▶ Lembra do problema do troco?
- ▶ Um algoritmo guloso sempre chegava ao resultado.
- ▶ Mas a solução nem sempre era ótima.

Algoritmos Gulosos

- ▶ Lembra do problema do troco?
- ▶ Um algoritmo guloso sempre chegava ao resultado.
- ▶ Mas a solução nem sempre era ótima.
- ▶ O resultado dependia dos valores das moedas.

Algoritmos Gulosos

- ▶ Algoritmos gulosos são elegantes e intuitivos.

Algoritmos Gulosos

- ▶ Algoritmos gulosos são elegantes e intuitivos.
- ▶ Se você tiver certeza que seu problema se encaixa numa solução gulosa, use-a!

Algoritmos Gulosos

- ▶ Algoritmos gulosos são elegantes e intuitivos.
- ▶ Se você tiver certeza que seu problema se encaixa numa solução gulosa, use-a!
- ▶ Pra ter certeza, no entanto, é preciso provar.
 - ▶ Introduction to Algorithms – Cormen.

Algoritmos Gulosos

- ▶ **Então:** algoritmos gulosos fazem, repetidamente, as melhores escolhas locais e ignoram o efeito que isso pode ter no futuro.

Algoritmos Gulosos

- ▶ **Então:** algoritmos gulosos fazem, repetidamente, as melhores escolhas locais e ignoram o efeito que isso pode ter no futuro.
- ▶ De certa forma, algoritmos gulosos podem ser vistos como uma subclasse especial de PD.

Algoritmos Gulosos

- ▶ Problemas que cabem uma solução gulosa, em geral, possuem as seguintes 2 características.

Algoritmos Gulosos

- ▶ Problemas que cabem uma solução gulosa, em geral, possuem as seguintes 2 características.
- ▶ **Sub-estrutura ótima:** solução ótima do *problema* engloba soluções ótimas dos *subproblemas*.

Algoritmos Gulosos

- ▶ Problemas que cabem uma solução gulosa, em geral, possuem as seguintes 2 características.
- ▶ **Sub-estrutura ótima:** solução ótima do *problema* engloba soluções ótimas dos *subproblemas*.
- ▶ **Propriedade de escolha gulosa:** soluções *ótimas locais* (e gulosas) levam a solução *ótima global*.

Algoritmos Gulosos

- ▶ Vamos ver dois exemplos em grafos que usam soluções gulosas.

Algoritmos Gulosos

- ▶ Vamos ver dois exemplos em grafos que usam soluções gulosas.
- ▶ O primeiro é a Minimum Spanning Tree (MST).
 - ▶ Encontra a árvore de menor peso em um grafo.

Algoritmos Gulosos

- ▶ Vamos ver dois exemplos em grafos que usam soluções gulosas.
- ▶ O primeiro é a Minimum Spanning Tree (MST).
 - ▶ Encontra a árvore de menor peso em um grafo.
- ▶ O segundo é o algoritmo de Dijkstra.
 - ▶ Encontra o menor caminho de um vértice a todos os outros no grafo.



Exemplos

Minimum Spanning Tree(MST)

Algoritmos Gulosos - MST

- ▶ Dado um grafo $G = (V, E)$, retorne uma árvore que tenha o menor peso possível.
 - ▶ Aliás, o que é uma árvore?

Algoritmos Gulosos - MST

- ▶ Dado um grafo $G = (V, E)$, retorne uma árvore que tenha o menor peso possível.
 - ▶ Aliás, o que é uma árvore?
- ▶ O uso de MST é muito comum em serviços de telecomunicação, pois ajudam a decidir a forma mais barata de interligar regiões.

Algoritmos Gulosos - MST

- ▶ Há dois algoritmos famosos que resolvem o problema de MST.
 - ▶ Algoritmo de Kruskal.
 - ▶ Algoritmo de Prim.

Algoritmos Gulosos - MST

- ▶ Há dois algoritmos famosos que resolvem o problema de MST.
 - ▶ Algoritmo de Kruskal.
 - ▶ Algoritmo de Prim.
- ▶ Ambos chegam no resultado ótimo usando escolhas gulosas.

Algoritmos Gulosos - MST

- ▶ Há dois algoritmos famosos que resolvem o problema de MST.
 - ▶ Algoritmo de Kruskal.
 - ▶ Algoritmo de Prim.
- ▶ Ambos chegam no resultado ótimo usando escolhas gulosas.
- ▶ Mas cada um tem sua particularidade.

Algoritmos Gulosos - Kruskal

- ▶ O algoritmo de Kruskal se baseia nas arestas.

Algoritmos Gulosos - Kruskal

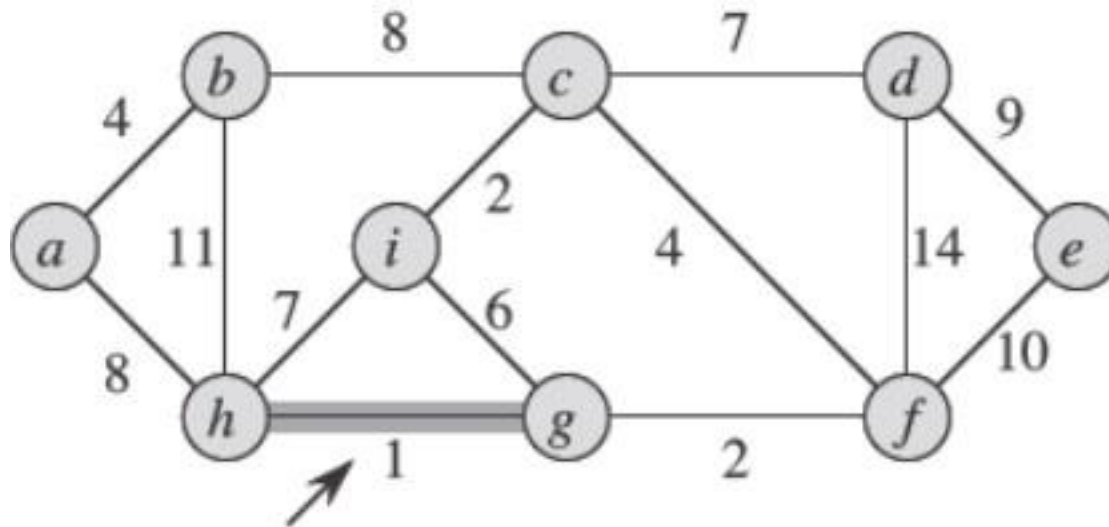
- ▶ O algoritmo de Kruskal se baseia nas arestas.
- ▶ Ele basicamente as ordena por menor peso e faz as escolhas tomando o cuidado de não formar ciclos.

Algoritmos Gulosos - Kruskal

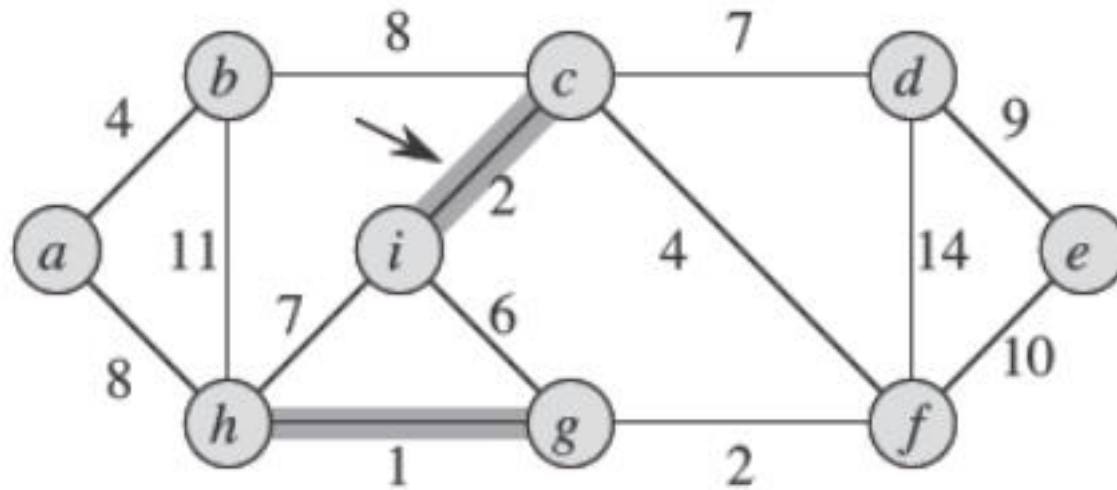
MST-KRUSKAL(G, w)

```
1   $A = \emptyset$ 
2  for each vertex  $v \in G.V$ 
3      MAKE-SET( $v$ )
4  sort the edges of  $G.E$  into nondecreasing order by weight  $w$ 
5  for each edge  $(u, v) \in G.E$ , taken in nondecreasing order by weight
6      if FIND-SET( $u$ )  $\neq$  FIND-SET( $v$ )
7           $A = A \cup \{(u, v)\}$ 
8          UNION( $u, v$ )
9  return  $A$ 
```

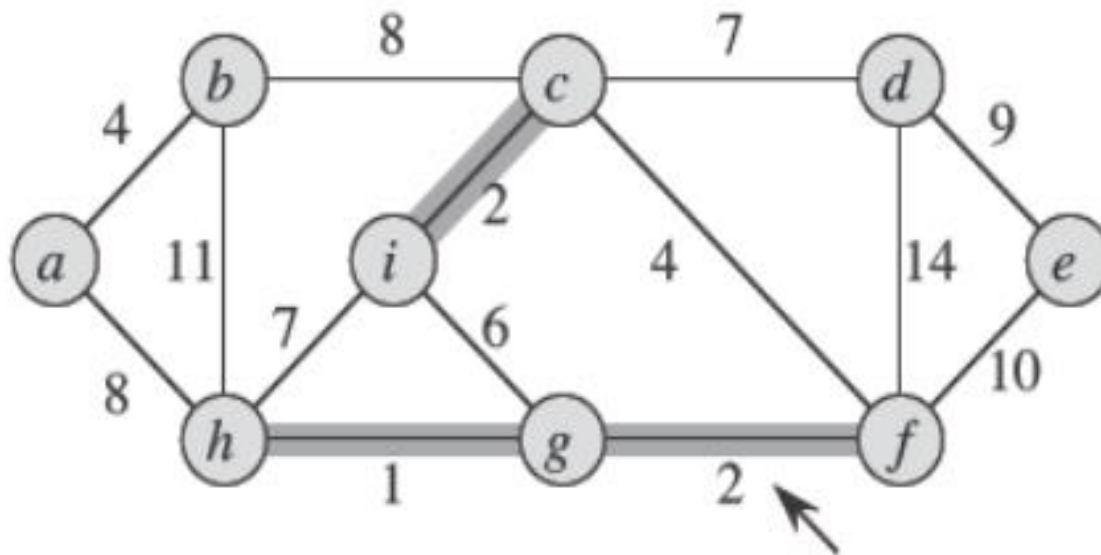
Algoritmos Gulosos - Kruskal



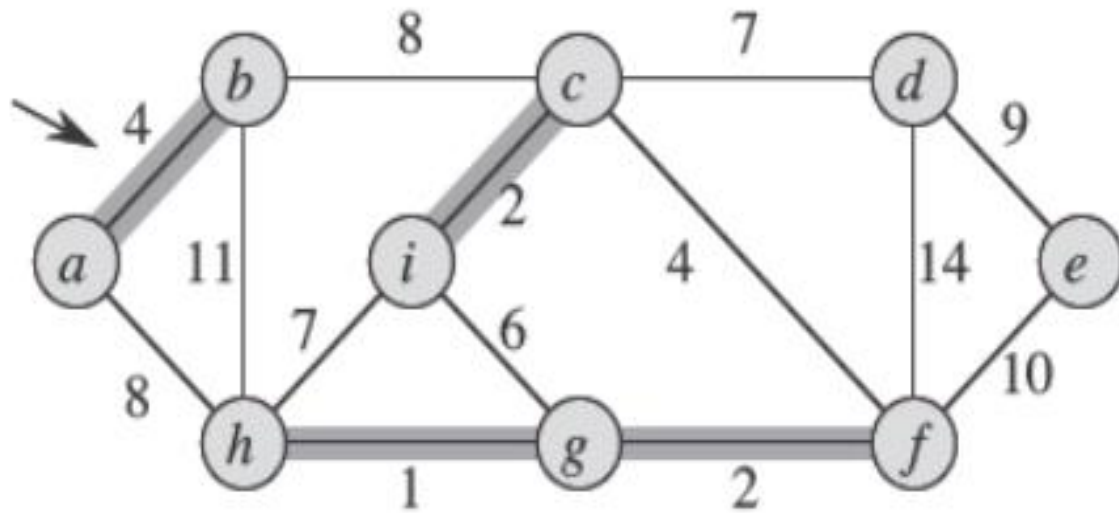
Algoritmos Gulosos - Kruskal



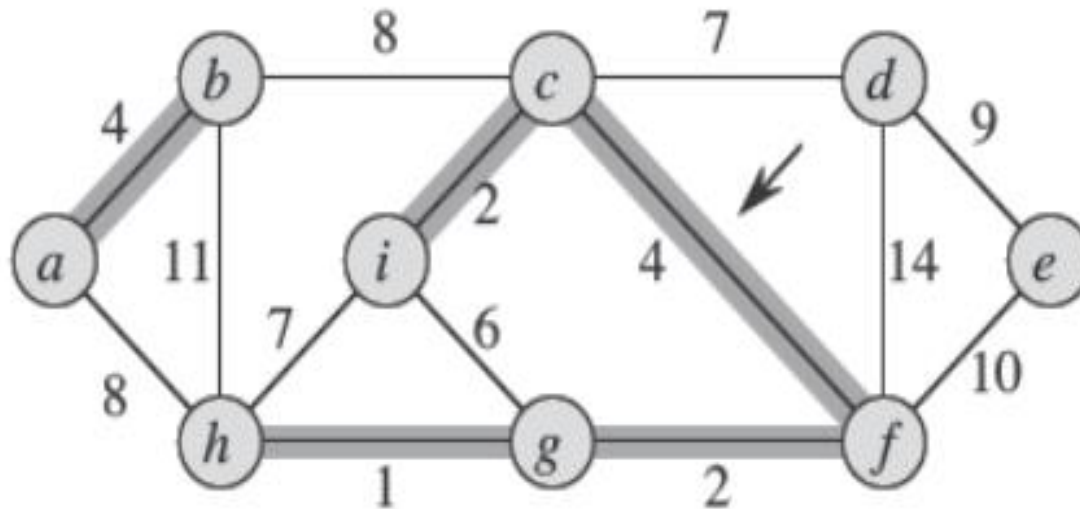
Algoritmos Gulosos - Kruskal



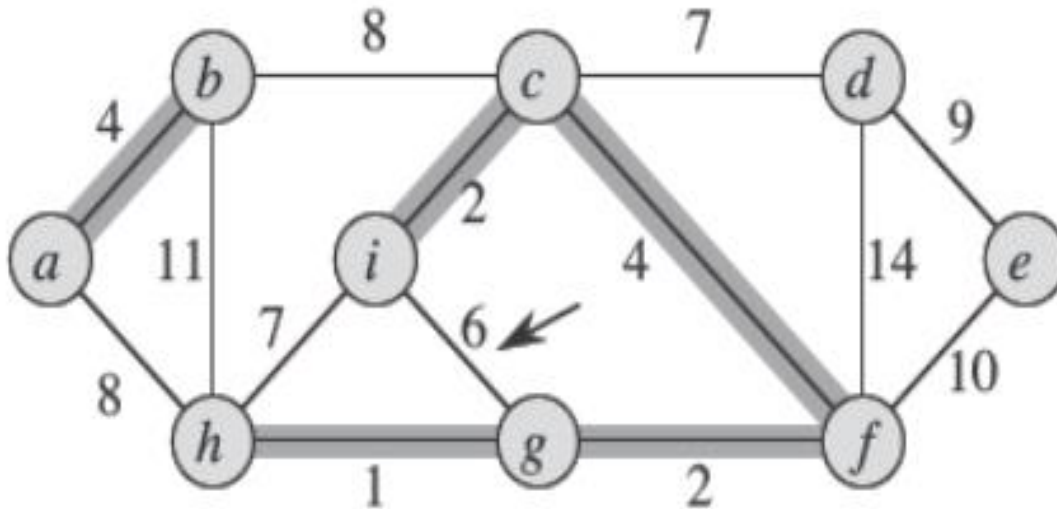
Algoritmos Gulosos - Kruskal



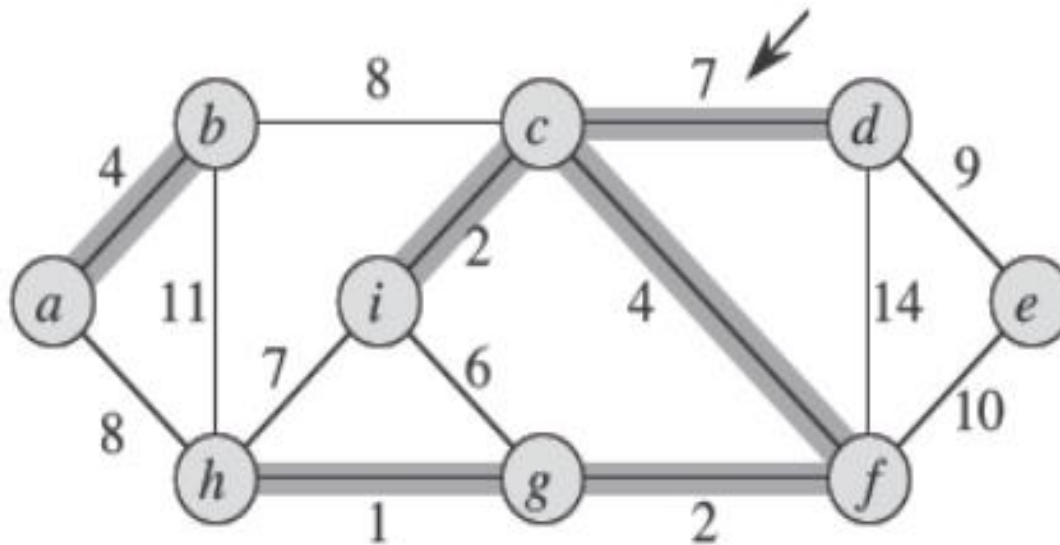
Algoritmos Gulosos - Kruskal



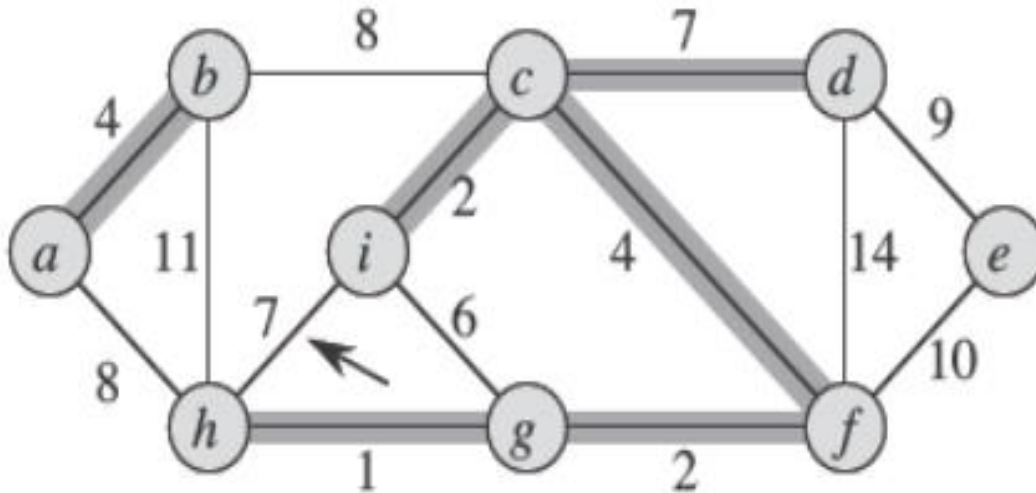
Algoritmos Gulosos - Kruskal



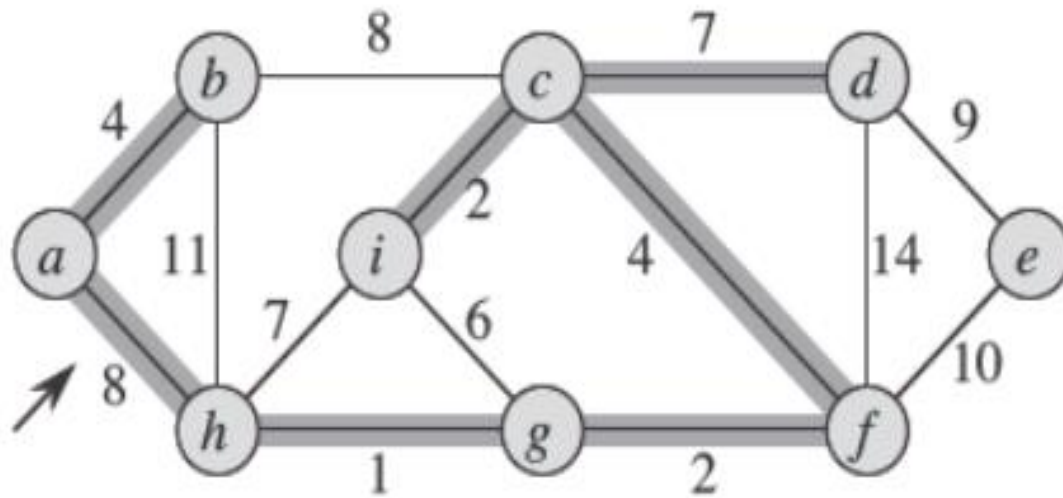
Algoritmos Gulosos - Kruskal



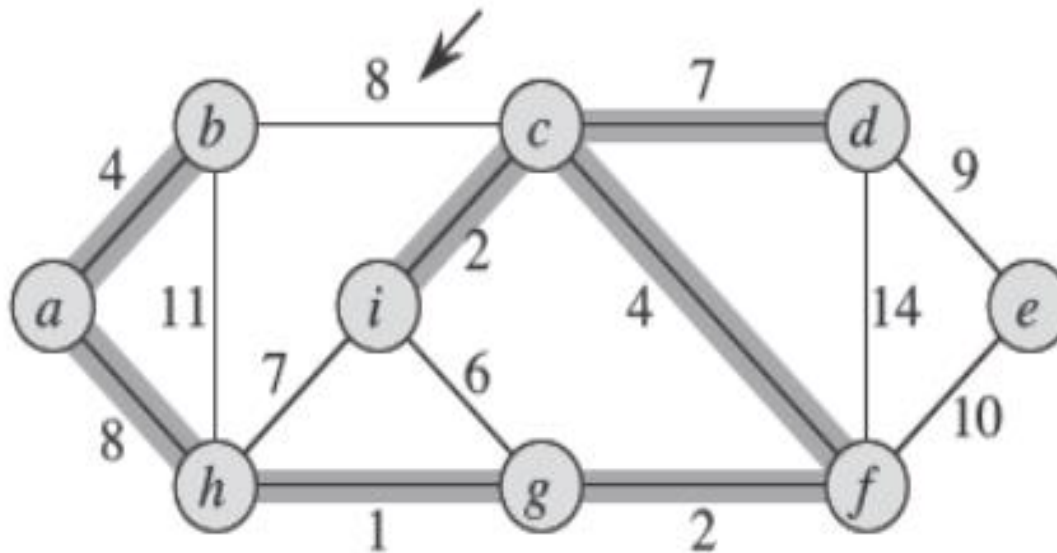
Algoritmos Gulosos - Kruskal



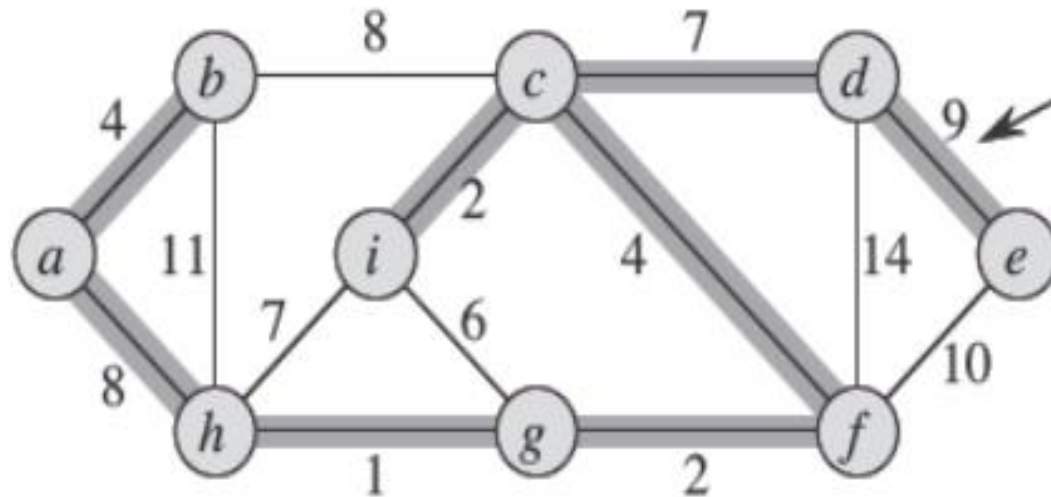
Algoritmos Gulosos - Kruskal



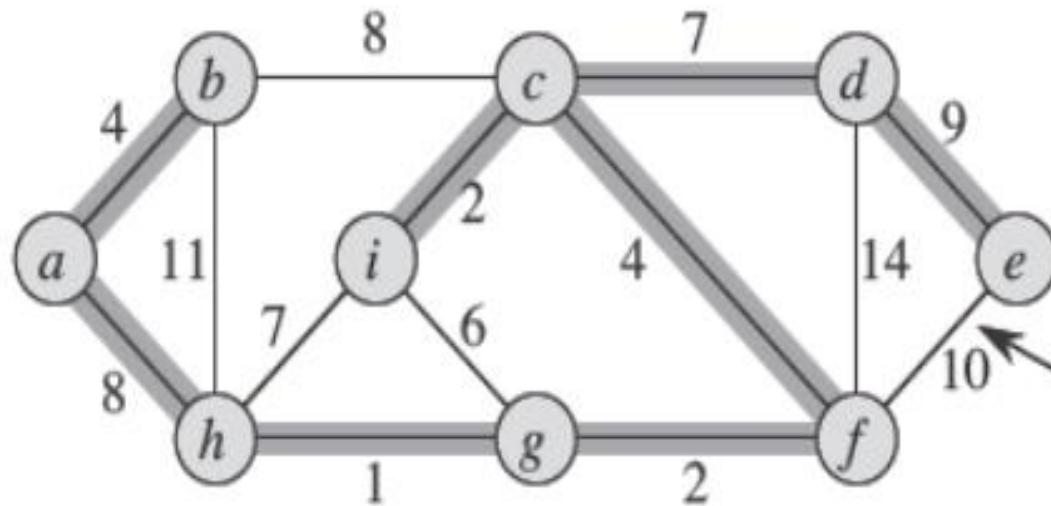
Algoritmos Gulosos - Kruskal



Algoritmos Gulosos - Kruskal



Algoritmos Gulosos - Kruskal



Algoritmos Gulosos - Prim

- ▶ O algoritmo de Prim se baseia nos vértices.

Algoritmos Gulosos - Prim

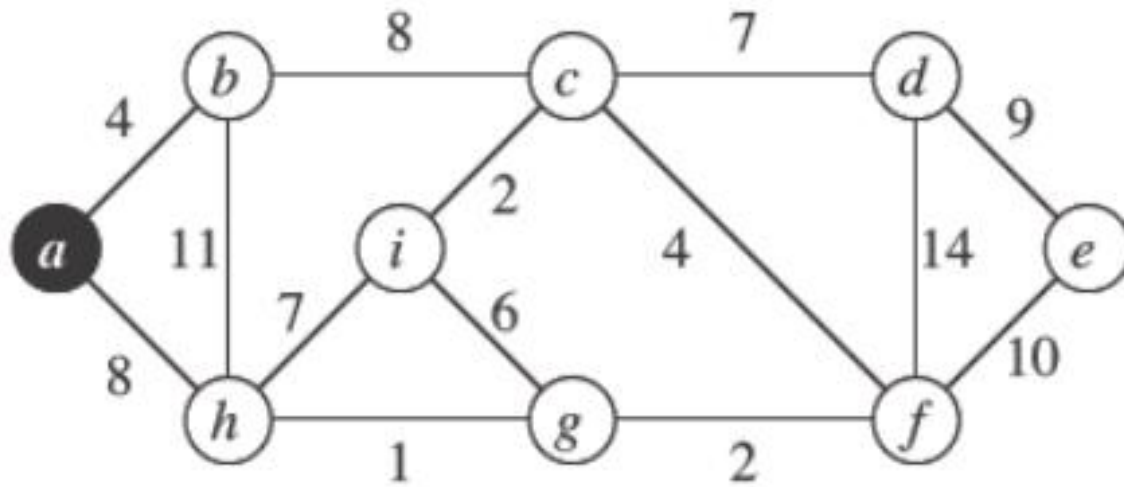
- ▶ O algoritmo de Prim se baseia nos vértices.
- ▶ Dado um vértice de início, o algoritmo cresce a árvore a partir daquele vértice, escolhendo sempre as arestas de menor peso.

Algoritmos Gulosos - Prim

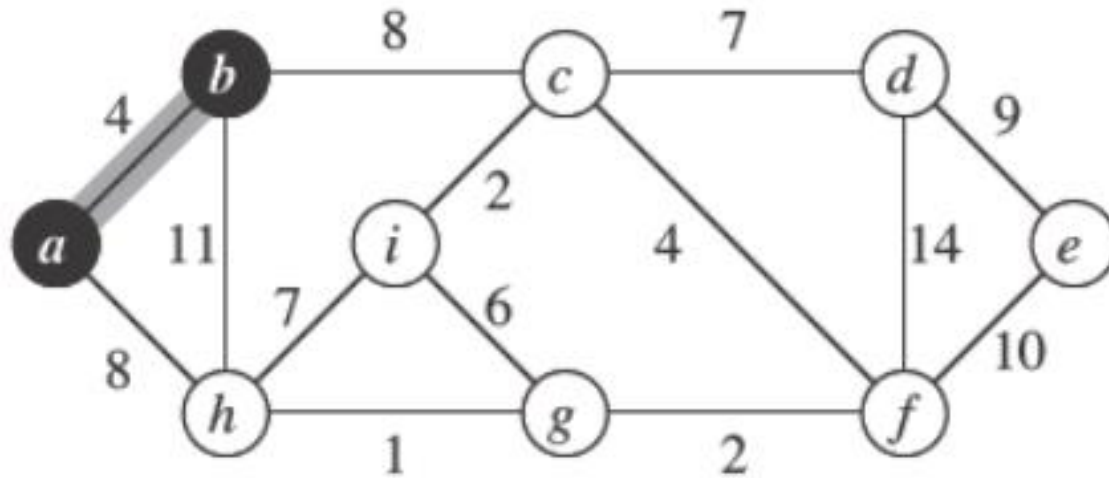
MST-PRIM(G, w, r)

```
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 
```

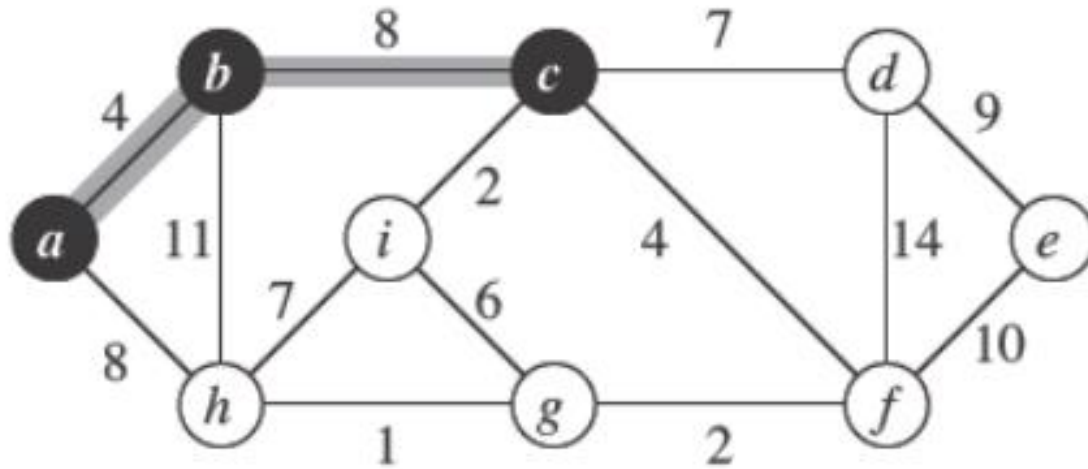
Algoritmos Gulosos - Prim



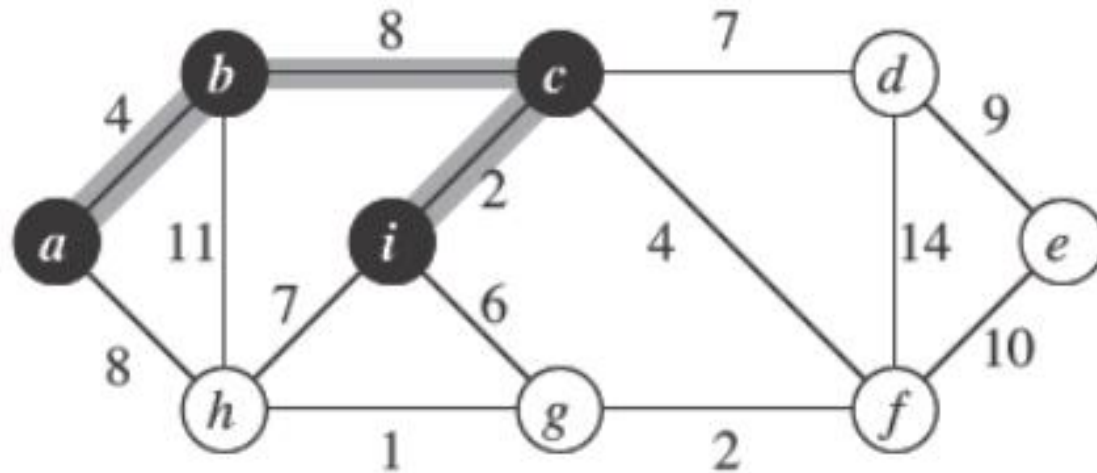
Algoritmos Gulosos - Prim



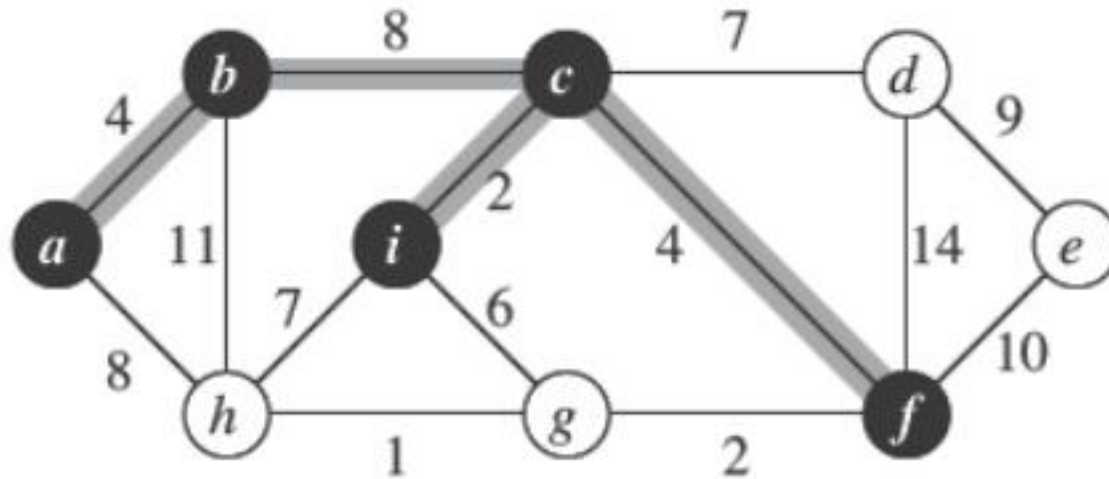
Algoritmos Gulosos - Prim



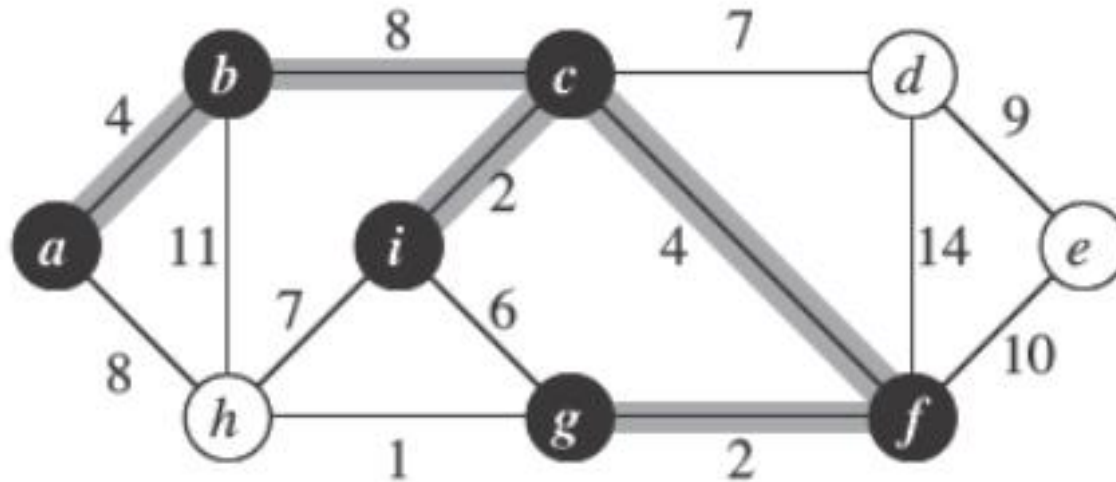
Algoritmos Gulosos - Prim



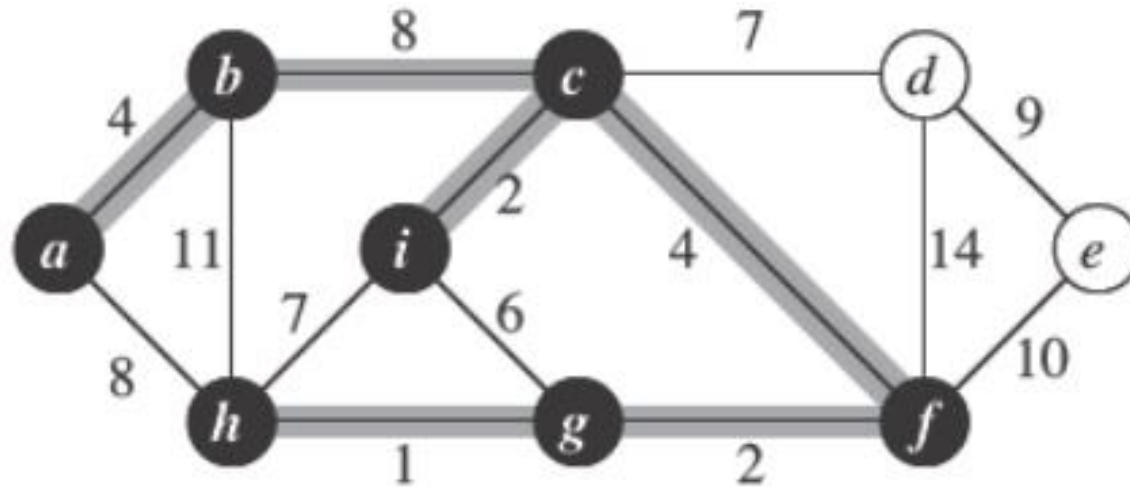
Algoritmos Gulosos - Prim



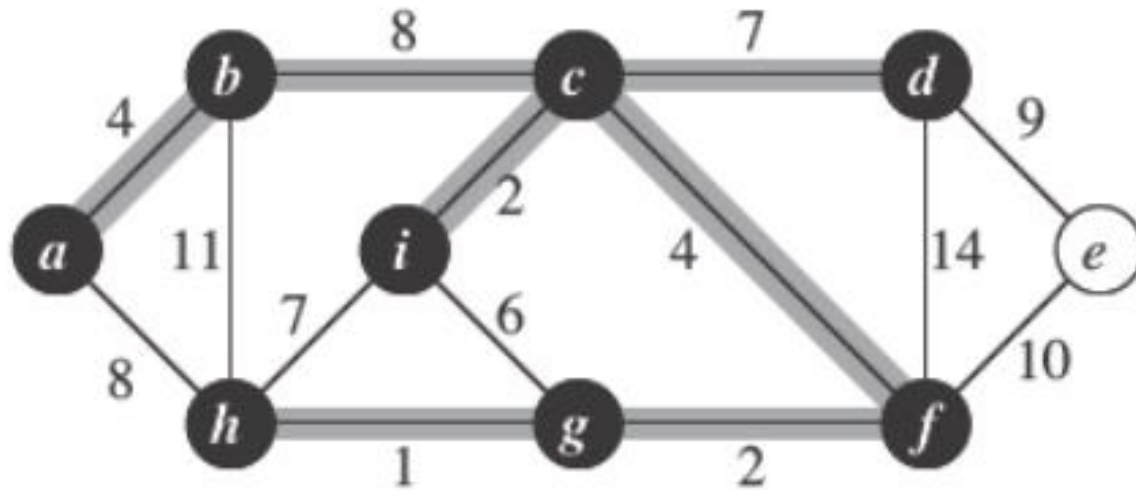
Algoritmos Gulosos - Prim



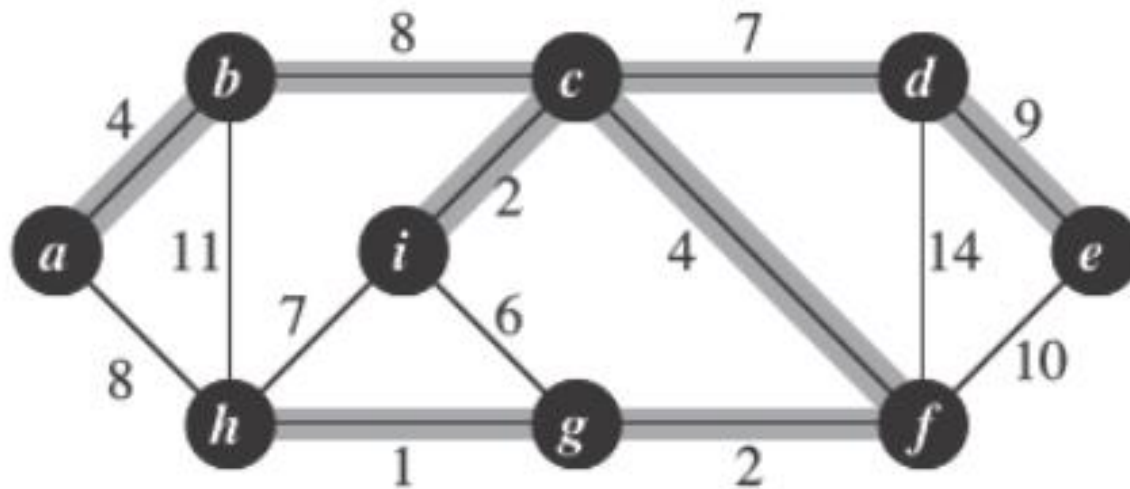
Algoritmos Gulosos - Prim



Algoritmos Gulosos - Prim



Algoritmos Gulosos - Prim



Algoritmos Gulosos - MST

- ▶ Qual é a complexidade das duas soluções?

Algoritmos Gulosos - MST

- ▶ Qual é a complexidade das duas soluções?
- ▶ Existe uma melhor escolha entre os dois algoritmos?



Algoritmo de Dijkstra



Algoritmos Gulosos - Dijkstra

- ▶ O algoritmo de Dijkstra é usado para calcular a menor distância de um vértice a todos os outros em um grafo direcionado sem pesos negativos.

Algoritmos Gulosos - Dijkstra

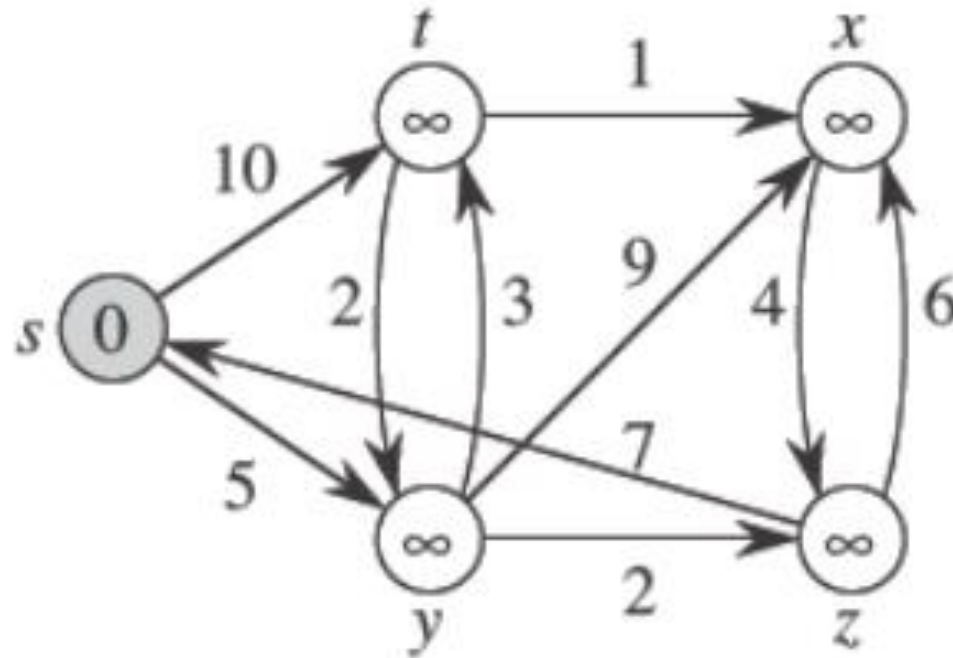
- ▶ O algoritmo de Dijkstra é usado para calcular a menor distância de um vértice a todos os outros em um grafo direcionado sem pesos negativos.
- ▶ Em grafos, é um dos algoritmos mais conhecidos.

Algoritmos Gulosos - Dijkstra

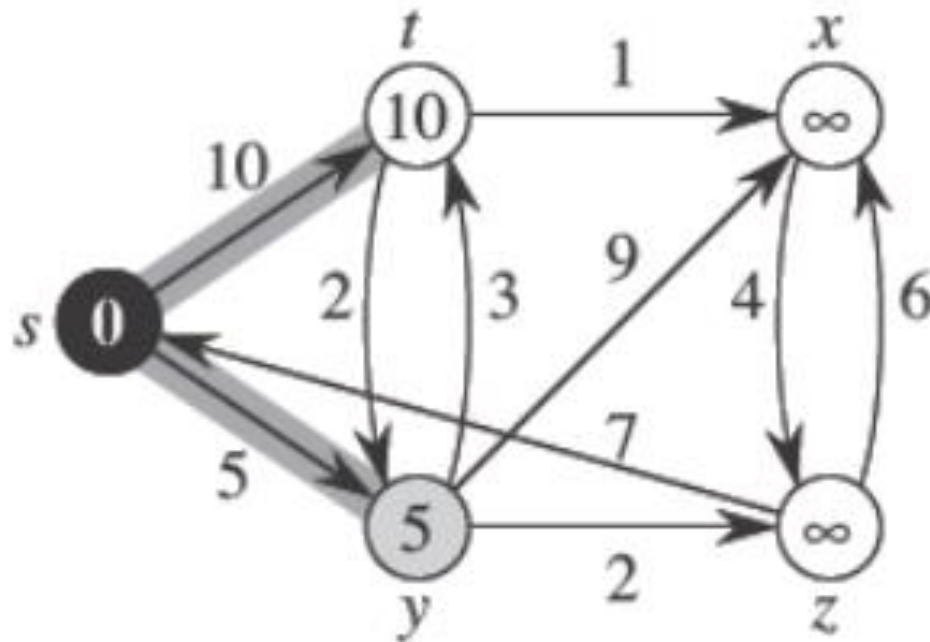
DIJKSTRA(G, w, s)

```
1  INITIALIZE-SINGLE-SOURCE( $G, s$ )
2   $S = \emptyset$ 
3   $Q = G.V$ 
4  while  $Q \neq \emptyset$ 
5       $u = \text{EXTRACT-MIN}(Q)$ 
6       $S = S \cup \{u\}$ 
7      for each vertex  $v \in G.Adj[u]$ 
8          RELAX( $u, v, w$ )
```

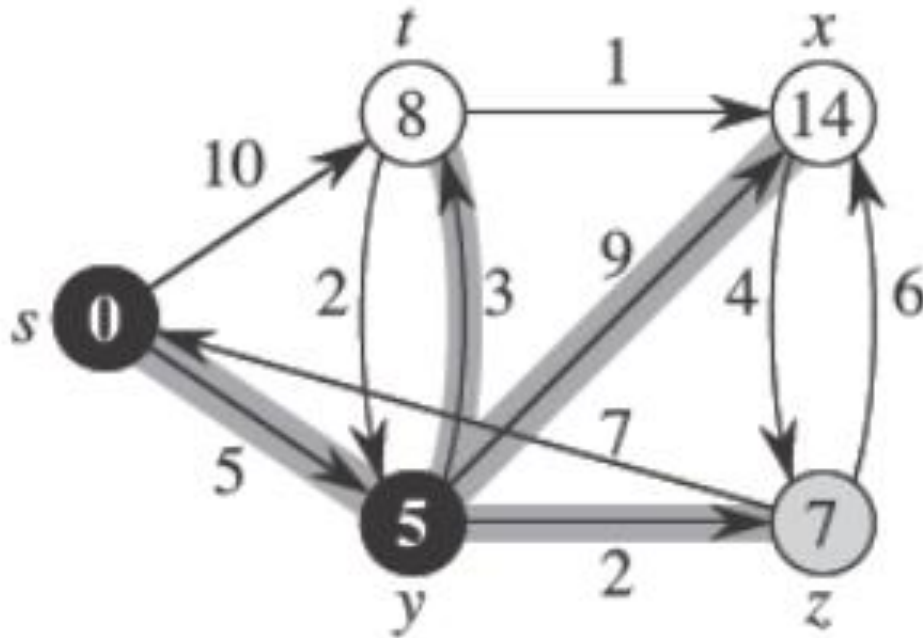

Algoritmos Gulosos - Dijkstra



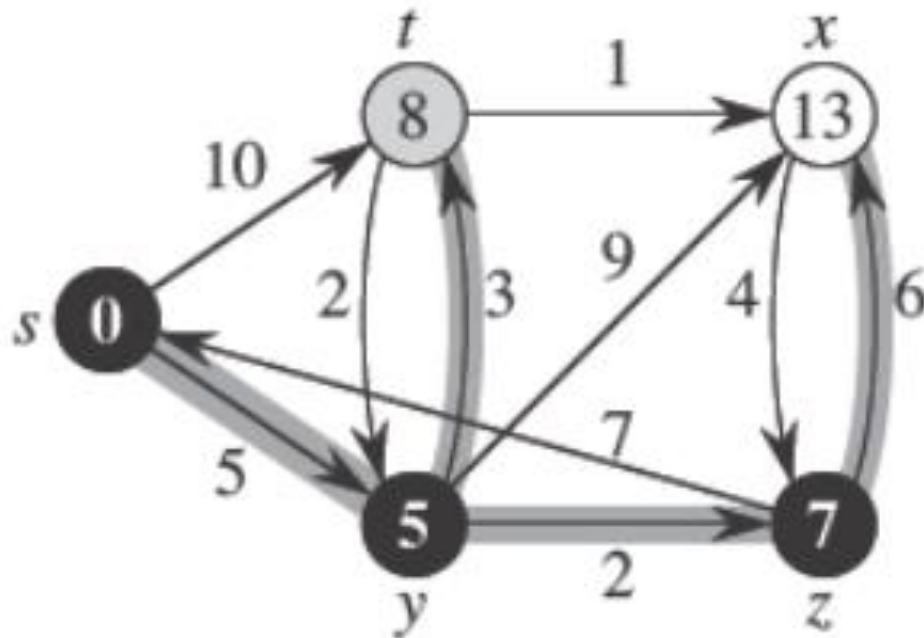
Algoritmos Gulosos - Dijkstra



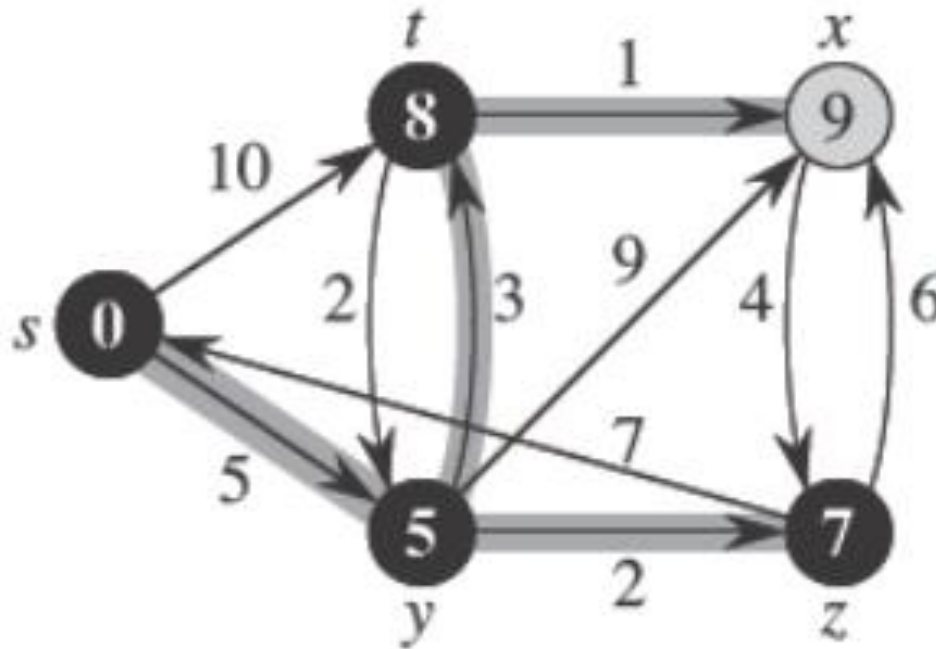
Algoritmos Gulosos - Dijkstra



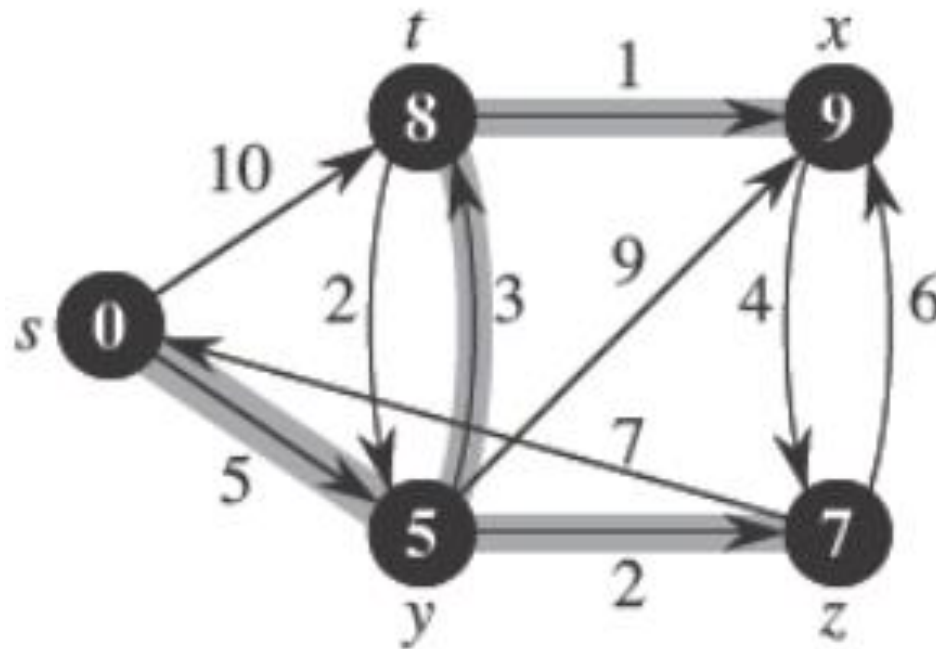
Algoritmos Gulosos - Dijkstra



Algoritmos Gulosos - Dijkstra



Algoritmos Gulosos - Dijkstra



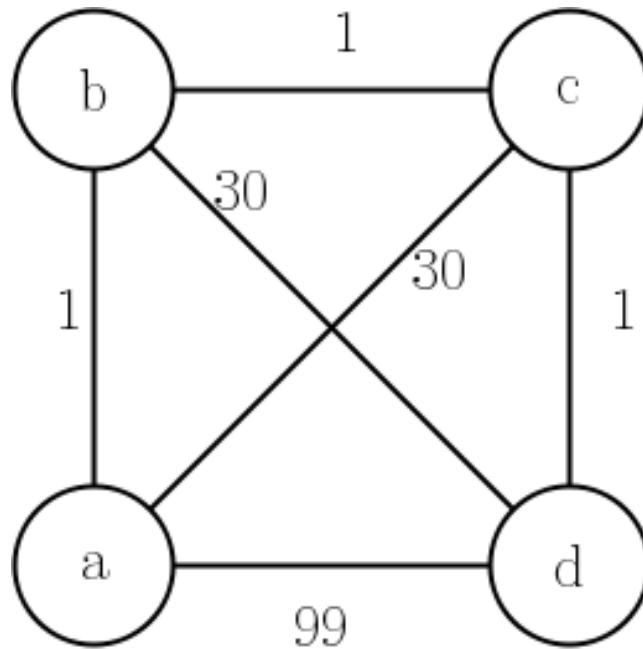
Algoritmos Gulosos - Dijkstra

- ▶ Posso usar Dijkstra pra resolver o problema do caixeiro viajante?!

Algoritmos Gulosos - Dijkstra

- ▶ Posso usar Dijkstra pra resolver o problema do caixeiro viajante?!
- ▶ **NÃO.** Eis a seguir um contra-exemplo.

Algoritmos Gulosos - Dijkstra



Algoritmos Gulosos - Dijkstra

- ▶ Qual é a complexidade do algoritmo de Dijkstra?

Algoritmos Gulosos - Dijkstra

- ▶ Qual é a complexidade do algoritmo de Dijkstra?
- ▶ Qual estrutura (já vista em aula) podemos usar pra melhorar a complexidade?



Resumo

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos sempre fazem as melhores escolhas do momento.

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos sempre fazem as melhores escolhas do momento.
- ▶ Com essas escolhas, esperamos chegar numa solução ótima.
 - ▶ Nem sempre acontece.

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos sempre fazem as melhores escolhas do momento.
- ▶ Com essas escolhas, esperamos chegar numa solução ótima.
 - ▶ Nem sempre acontece.
- ▶ Uma boa dica é sempre tentar achar contra-exemplos.

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos são elegantes e intuitivos.

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos são elegantes e intuitivos.
- ▶ Grande aplicabilidade em problemas de grafos.
 - ▶ Árvores mínimas.
 - ▶ Menor caminho.
 - ▶ Coloração de vértices.

Algoritmos Gulosos - Resumo

- ▶ Algoritmos gulosos são elegantes e intuitivos.
- ▶ Grande aplicabilidade em problemas de grafos.
 - ▶ Árvores mínimas.
 - ▶ Menor caminho.
 - ▶ Coloração de vértices.
- ▶ Conheça bem seu problema pra saber onde aplicar uma solução gulosa.



Perguntas?