FCT/Unesp – Presidente Prudente
Departamento de Matemática e Computação

Projeto e Análise de Algoritmos: Algoritmos Gulosos

Prof. Danilo Medeiros Eler
danilo.eler@unesp.br

Departamento de Matemática e Computação
Faculdade de Ciências e Tecnologia
Unesp – Universidade Estadual Paulista
Presidente Prudente/SP, Brasil

DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Algoritmos Gulosos

- Algoritmo guloso, ou ganancioso
  - Em inglês *greedy algorithms*
- Para resolver um problema ele escolhe o objeto mais "apetitoso" que vê pela frente
- A definição de "apetitoso" é estabelecida a priori
- O objeto escolhido passa a fazer parte da solução que o algoritmo constrói

unesp
DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Algoritmos Gulosos

- Podemos dizer que um algoritmo guloso é “míope”, pois toma as decisões com base nas informações disponíveis na iteração atual
  - Ele espera que a escolha de um ótimo local em cada etapa leve a um ótimo global
- Em alguns casos, algoritmos gulosos podem funcionar bem para problemas de otimização
  - Esses problemas não buscam encontrar uma solução, mas sim a melhor solução

Algoritmos Gulosos - Exemplo

- Suponha que tenhamos que fazer um algoritmo que devolva um determinada quantia de troco (*change-making problem*), usando o menor número de moedas possível
- Um algoritmo guloso para esse problema faria em cada passo
  - pegue a maior nota ou moeda possível que não ultrapasse o valor restante

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 289

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 189

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 89

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 39

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 14

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 4

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 3

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 2

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 1

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 50, 100
  - R$ 1,00 = 100 centavos
- Restante: 0

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- Restante: 289
- Se retirarmos a moeda de 50 centavos, o algoritmo ainda funcionaria?

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- Restante: 89

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- Restante: 14

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- Restante: 4

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- Restante: 0

Algoritmos Gulosos

- Exemplo R$ 2,89
- Moedas: 1, 5, 10, 25, 100
  - R$ 1,00 = 100 centavos
- O algoritmo guloso funciona bem para esse sistema monetário
- Ele ainda encontraria a melhor solução para outro sistema monetário?
  - Ex.: sistema fictício com moedas 1, 7 e 10

Algoritmos Gulosos

- Exemplo R$ 0,15
- Moedas: 1, 5, 10, 25, 100
  - 10 e 5
- Moedas: 1, 7 e 10
  - 10, 1, 1, 1, 1 e 1
  - É a melhor solução?

Algoritmos Gulosos

- Exemplo R$ 0,15
- Moedas: 1, 5, 10, 25, 100
  - 10 e 5
- Moedas: 1, 7 e 10
  - 10, 1, 1, 1, 1 e 1
  - É a melhor solução?
  - Não
  - A melhor solução seria: 7, 7 e 1

Algoritmos Gulosos

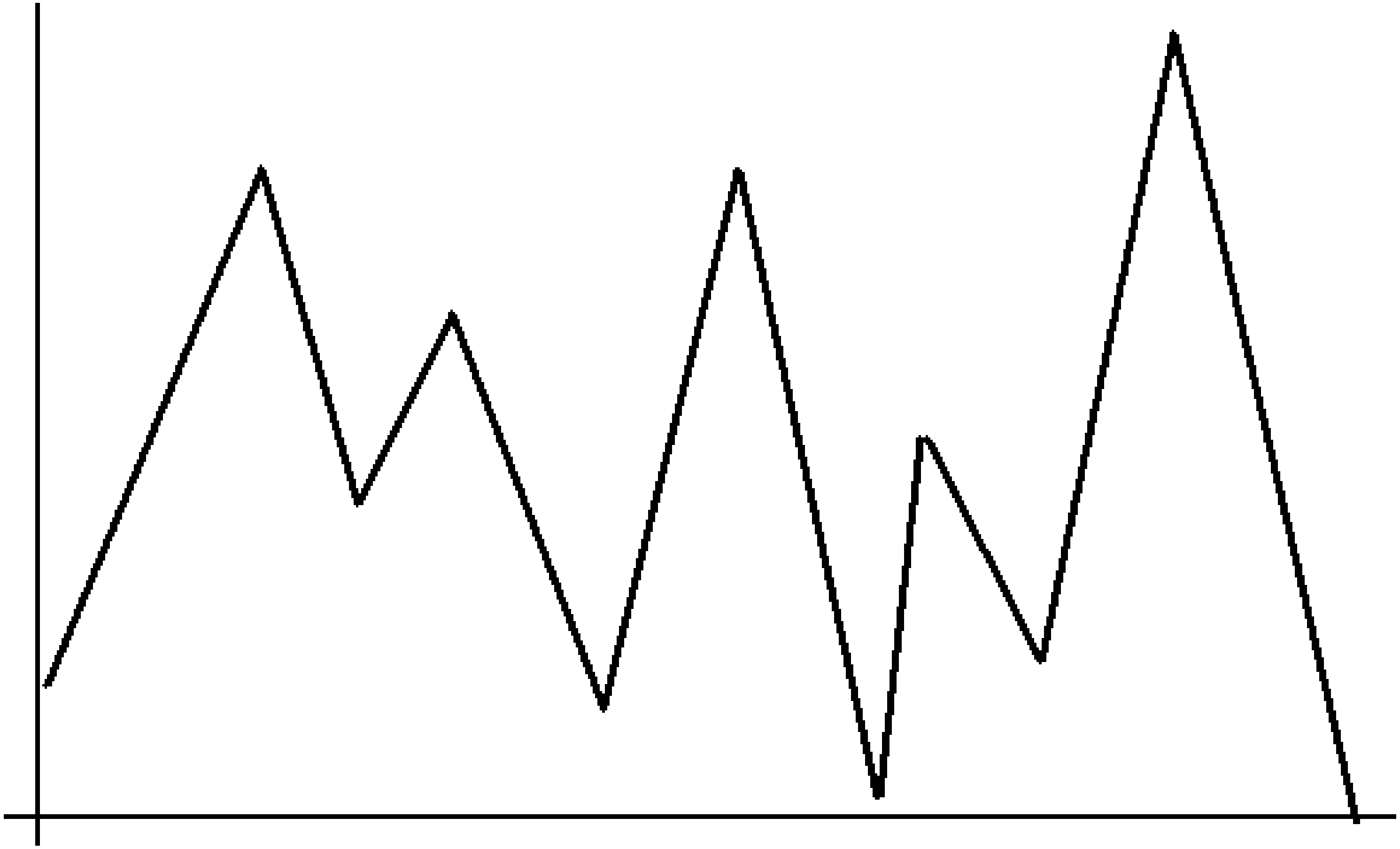
- Exemplo R$ 0,15
- Moedas: 1, 5, 10, 25, 100
  - 10 e 5
- Moedas: 1, 7 e 10
  - 10, 1, 1, 1, 1 e 1 **(ótimo local)**
  - É a melhor solução?
  - Não
  - A melhor solução seria: 7, 7 e 1 **(ótimo global)**

Algoritmos Gulosos

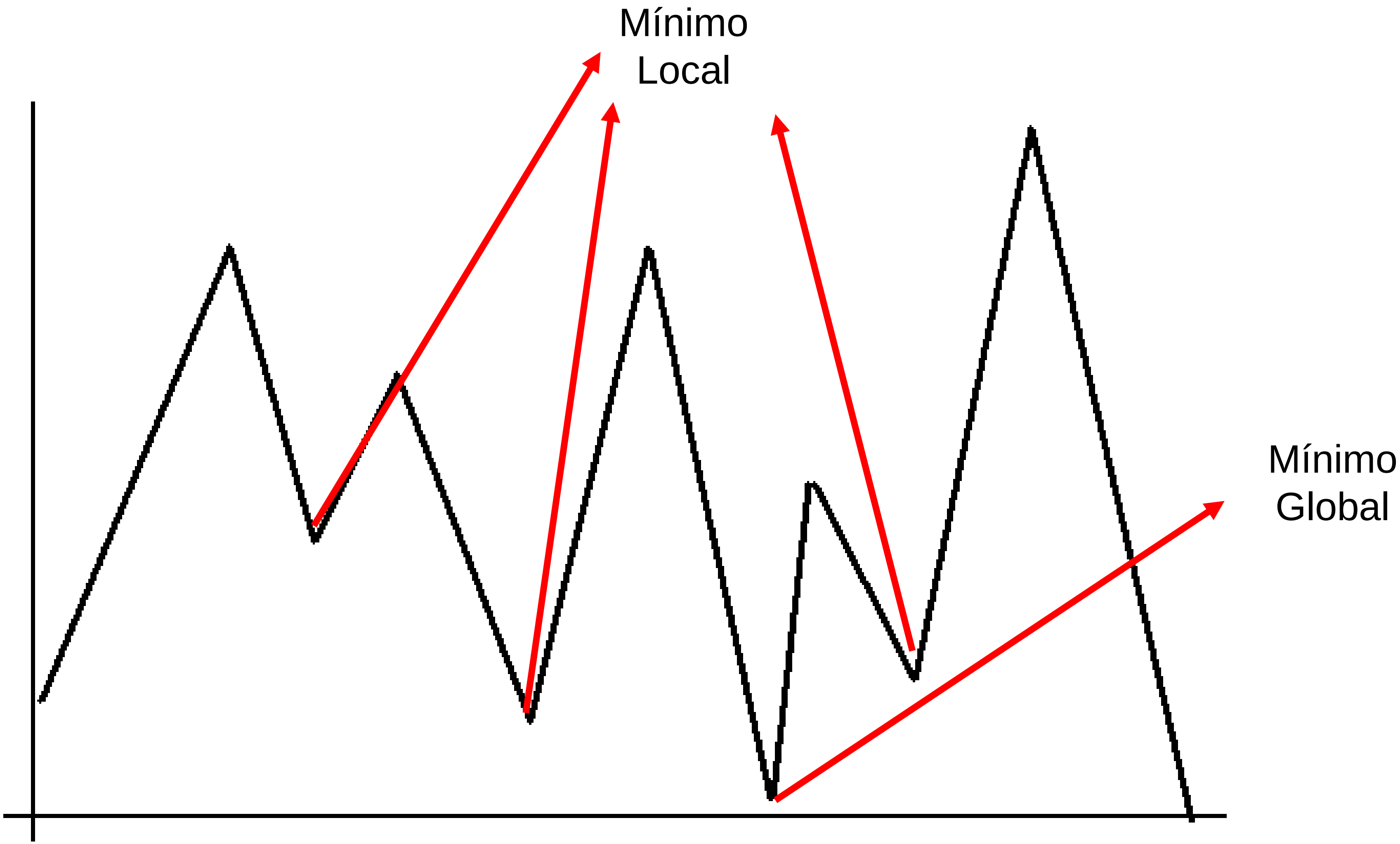
- Exemplo R$ 0,15
- Moedas: 1, 5, 10, 25, 100
  - 10 e 5
- Moedas: 1, 7 e 10
  - 10, 1, 1, 1, 1 e 1 **(ótimo local)**
  - É a melhor solução?
  - Não
  - A melhor solução seria: 7, 7 e 1 **(ótimo global)**

Algoritmos Gulosos

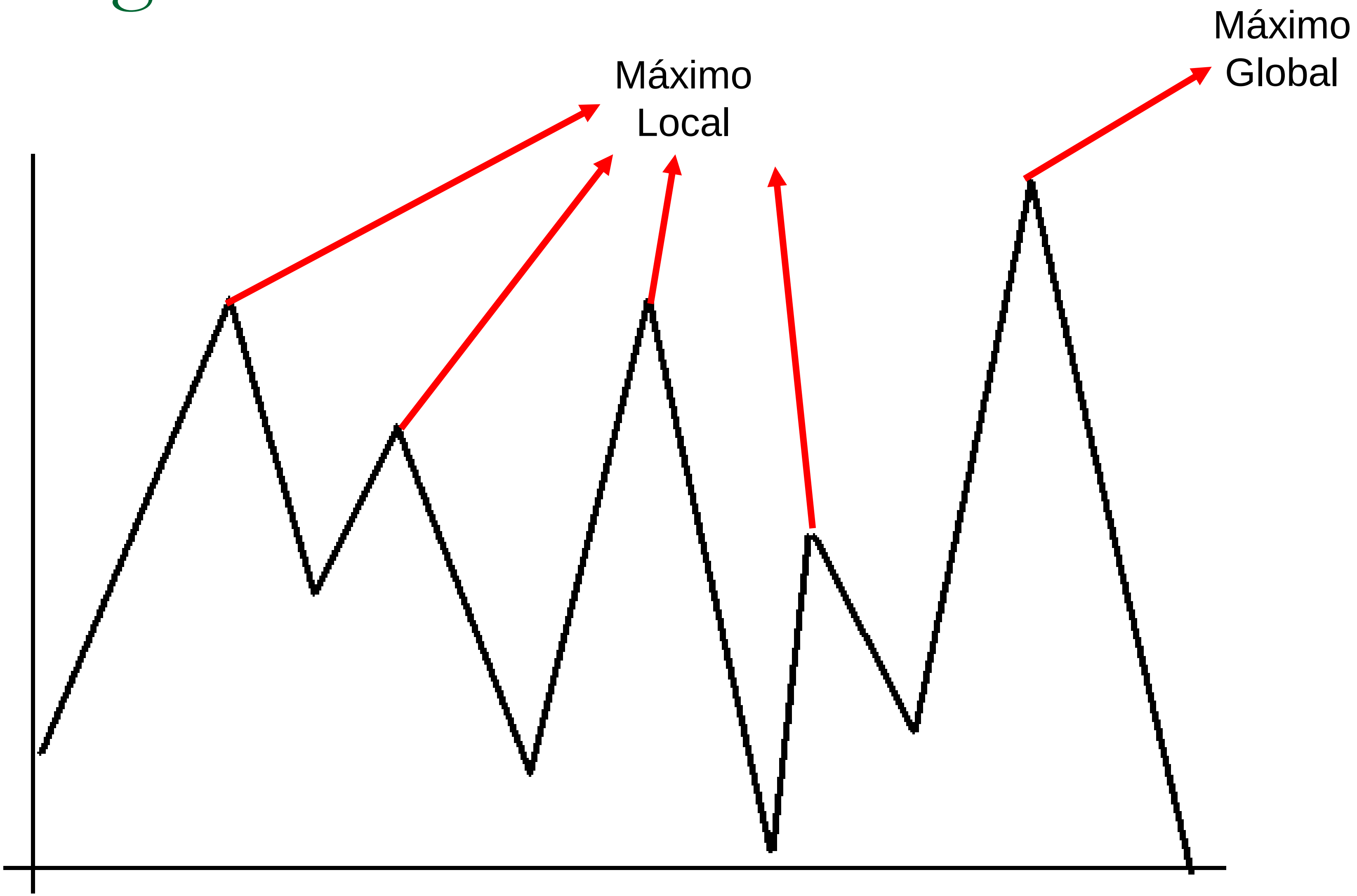
Possibilidades de troco



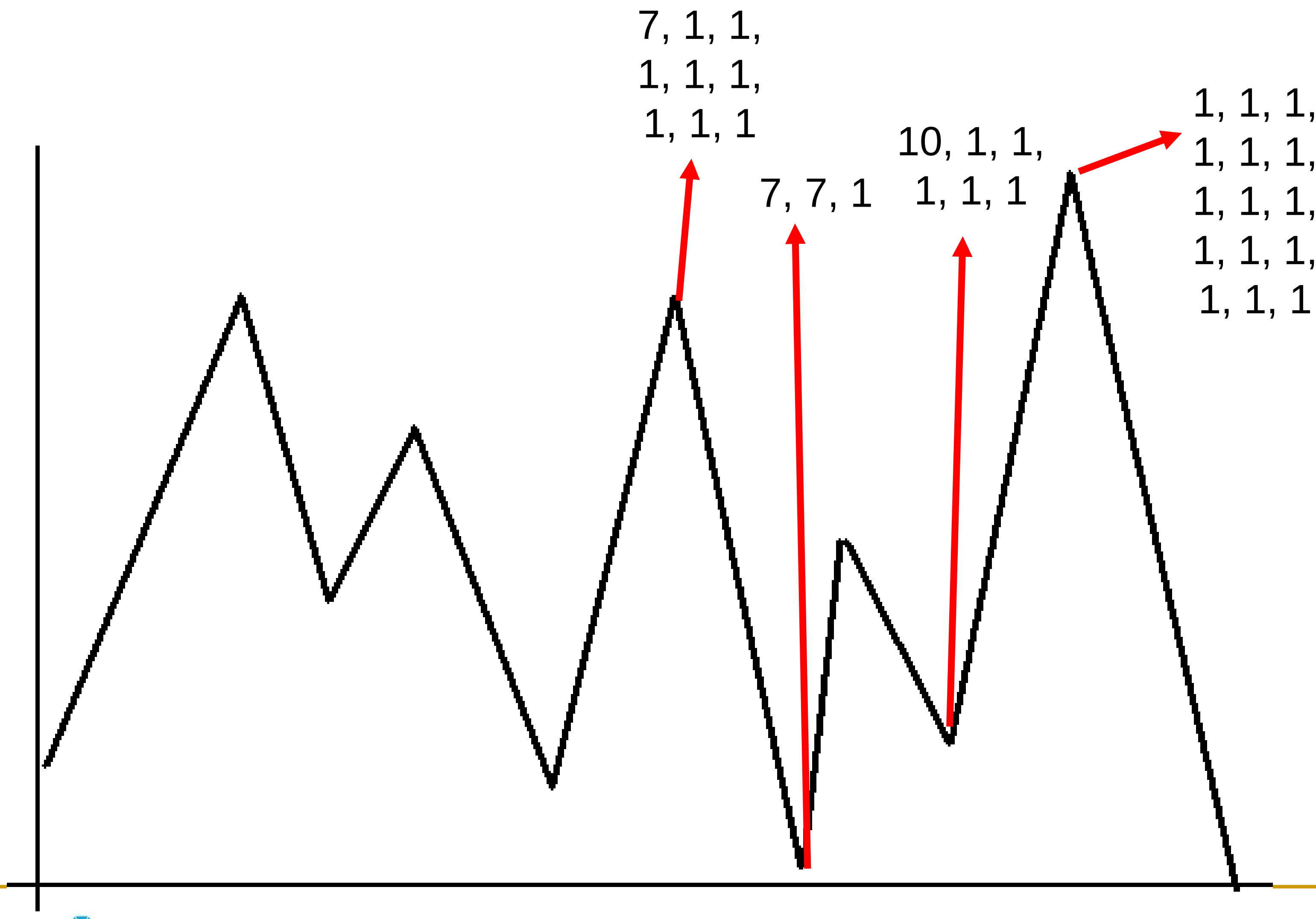
Algoritmos Gulosos



Algoritmos Gulosos



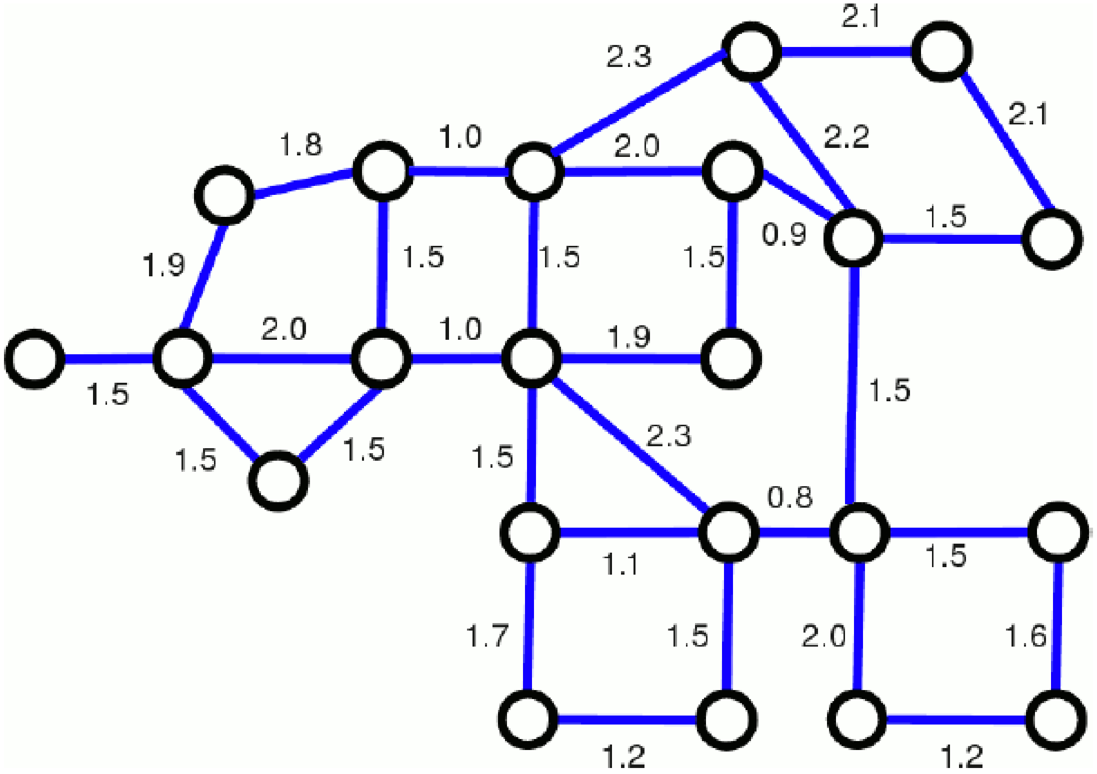
Algoritmos Gulosos



Outro Exemplo

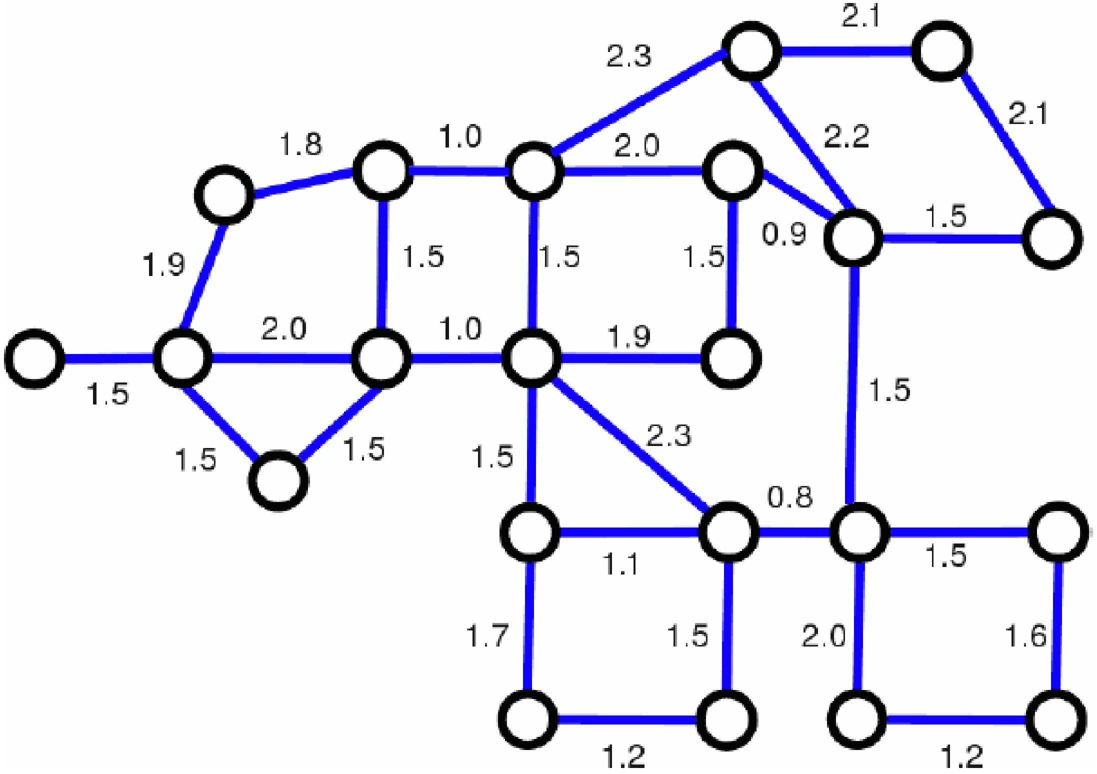
Ilhas e possíveis pontes

- Desejamos conectar todas as ilhas por pontes, com o menor custo?



Comunicação em uma empresa

- Como construir uma linha privada de comunicação entre os escritórios, com o menor custo?



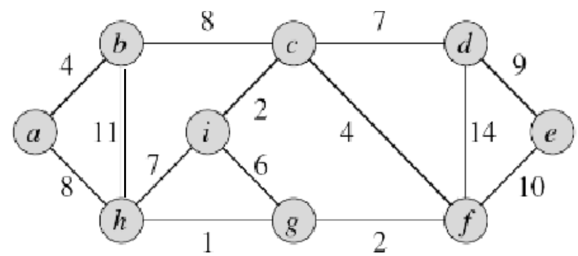
Árvore Geradora Mínima

Para cada aresta *(u,v)* pertencente a A, temos um **peso *w(u,v)*** especificando o custo para interconectar *(u,v)*. Então, desejamos encontrar um **subconjunto acíclico *X*** contido ou igual a A, que **conecte todos os vértices e cujo peso total seja minimizado**.

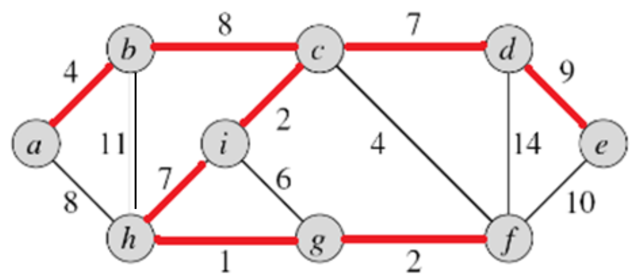
$$G = (V, A) \longrightarrow G' = (V, X)$$

$$\min \; w(X) = \sum_{(u,v) \in X} w(u,v)$$

$G = (V, A)$

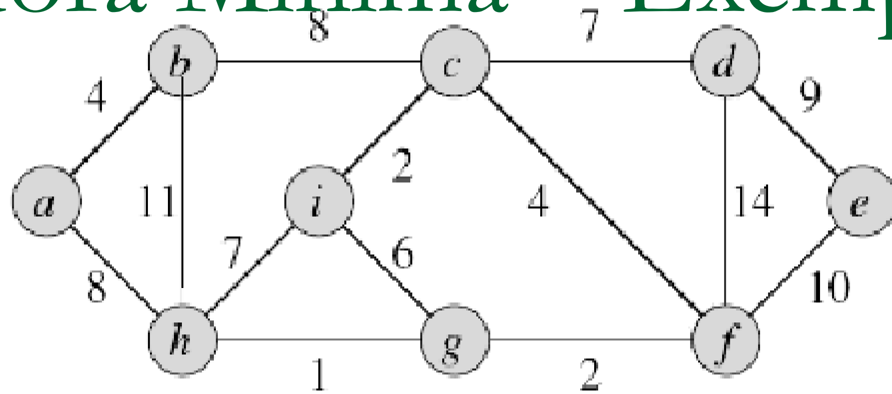


$G' = (V, X)$



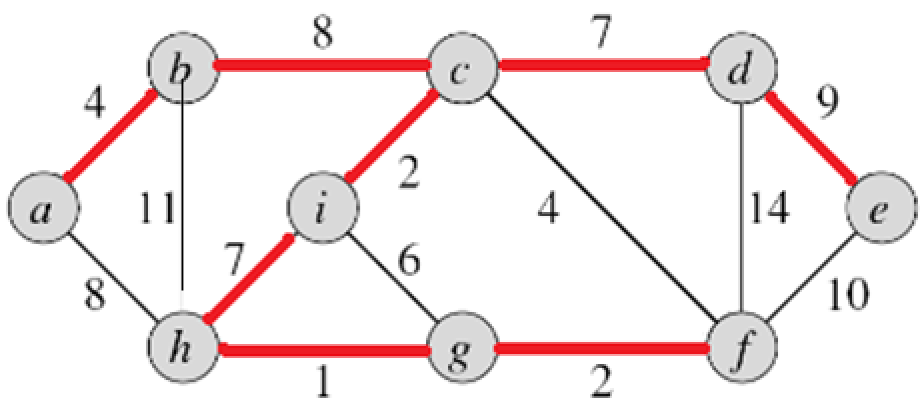
Árvore Geradora Mínima – Exemplo

$$G = (V, A)$$



$$\min \; w(X) = \sum_{(u,v)\in X} w(u,v) = \mathbf{40}$$

$$G' = (V, X)$$

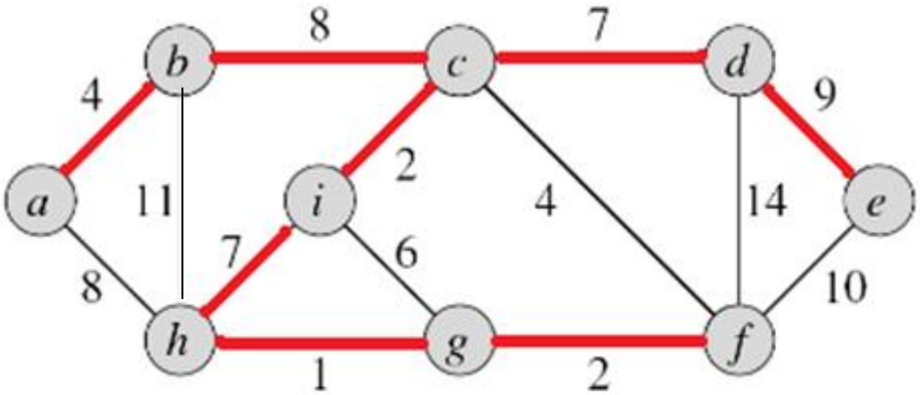


Não é a Árvore que tem a menor soma de pesos das arestas

Árvore Geradora Mínima – Exemplo

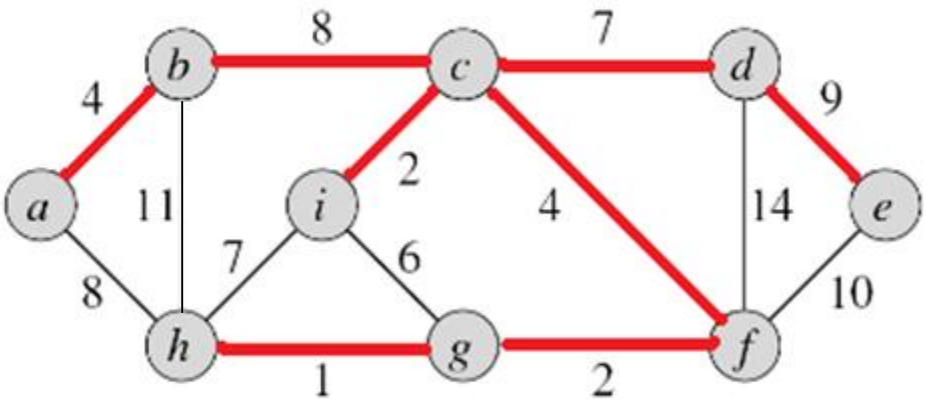
$$\min \; w(X) = \sum_{(u,v)\in X} w(u,v) = \mathbf{40}$$

$$G' = (V, X)$$



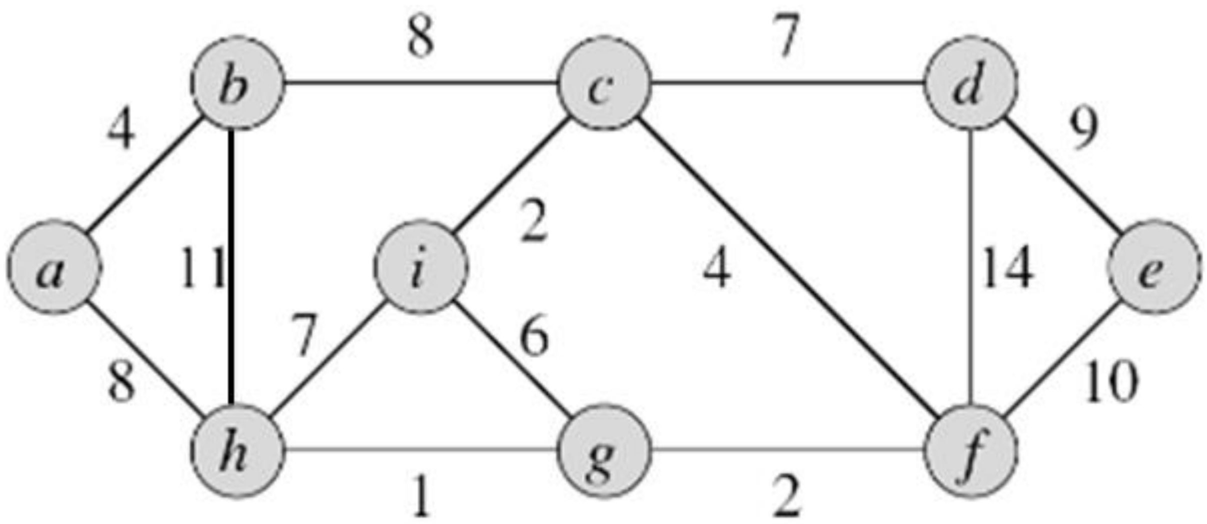
$$\min \; w(X) = \sum_{(u,v)\in X} w(u,v) = \mathbf{37}$$

$$G' = (V, X)$$



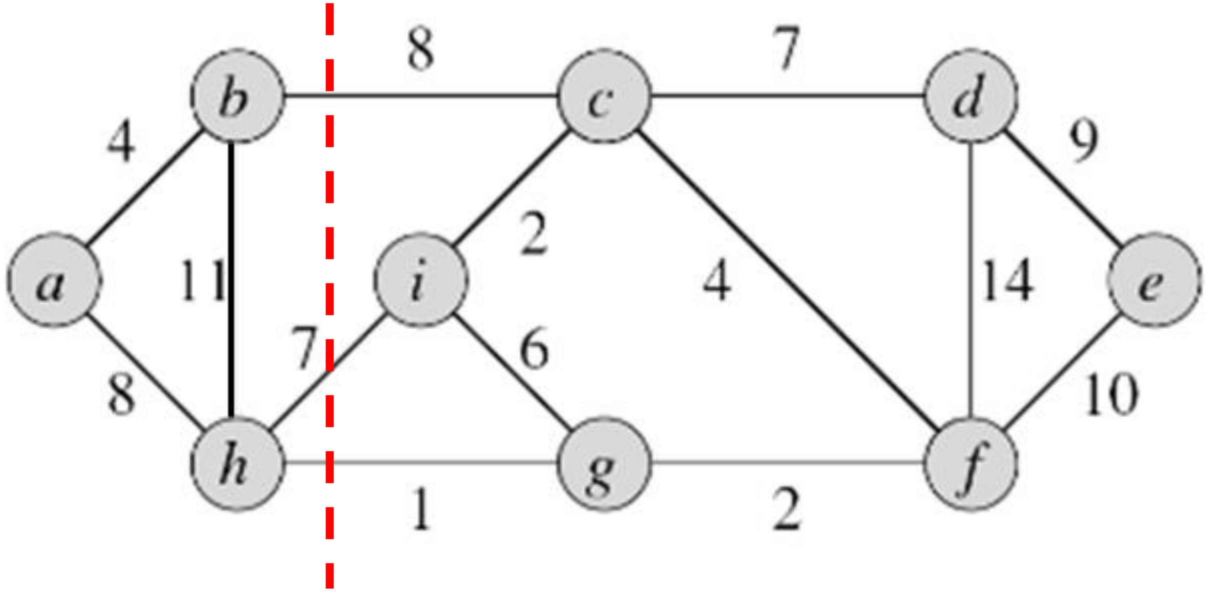
Árvore Geradora Mínima – Exemplo

- Aplicar um algoritmo guloso para o grafo abaixo



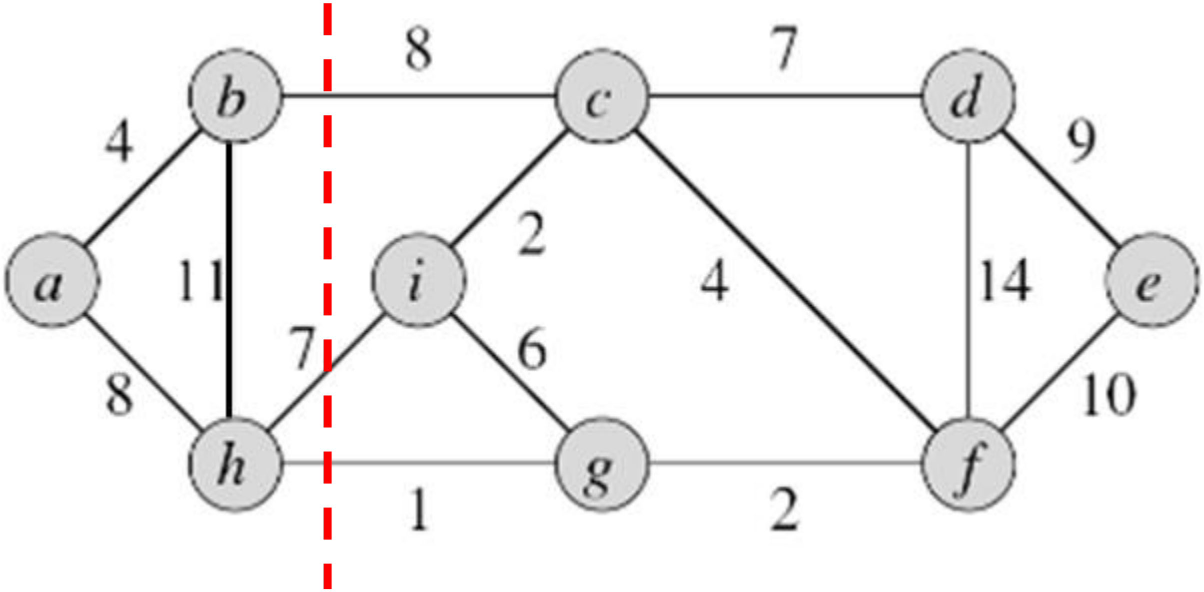
Exemplo de um Algoritmo Guloso

- Corte em Grafo
  - Separar em dois conjuntos S e V-S
    - S = {a, b, h}
    - V-S = {i, c, d, e, f, g}



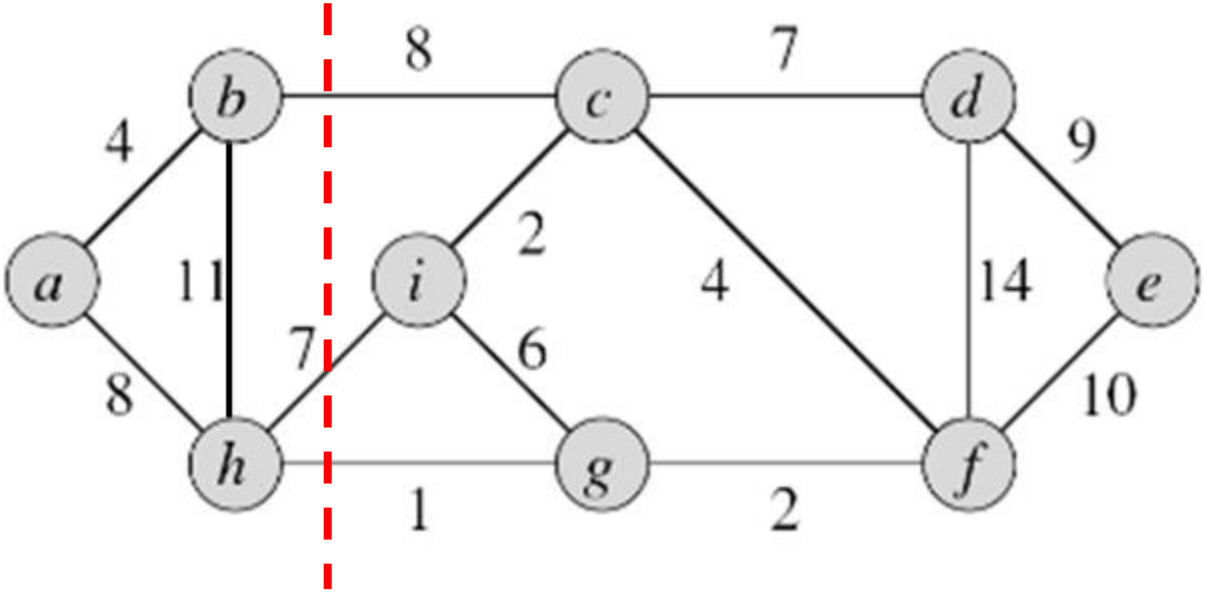
Exemplo de um Algoritmo Guloso

- Aresta Leve
  - Aresta de menor peso que cruza o corte
    - Aresta (h,g) com peso 1



Exemplo de um Algoritmo Guloso

- Aresta Leve
  - Aresta de menor peso que cruza o corte
    - Aresta (h,g) com peso 1

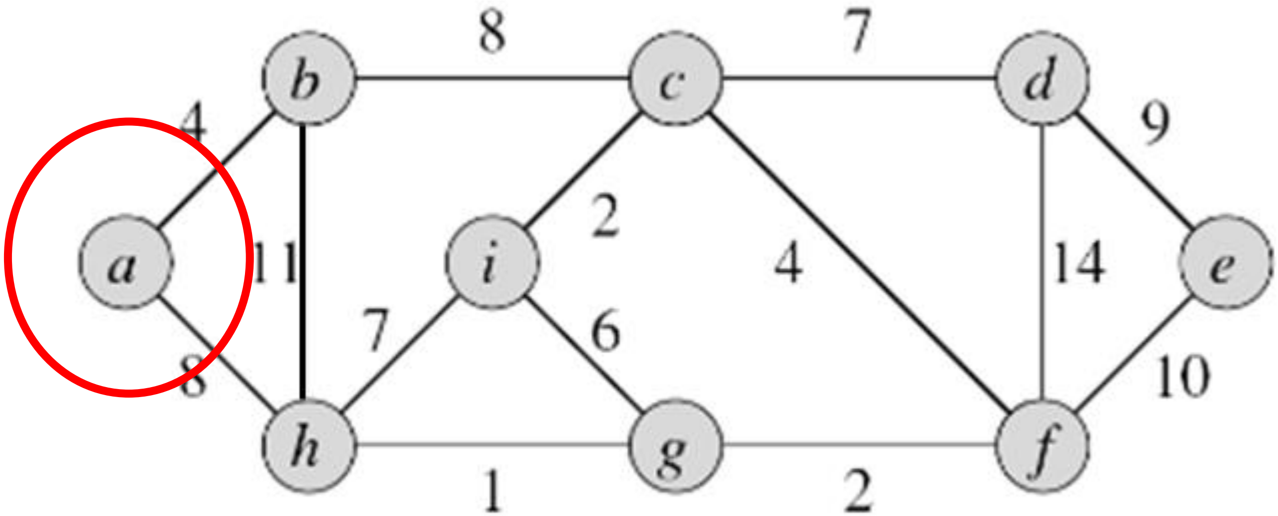


Exemplo de um Algoritmo Guloso

1. Selecionar um vértice inicial para o conjunto S
    1. Assim, um corte inicial é definido
2. Encontrar a aresta leve (i,j) que cruza o corte e adicionar como solução
3. Atualizar o conjunto S adicionando o vértice ‘i’ ou ‘j’, isto é, aquele que não pertencia ao conjunto S
    1. Assim, o corte foi atualizado para que as arestas do conjunto solução não cruzem o corte
4. Se V-S diferente de vazio, repetir a partir do <u>Passo 2</u>

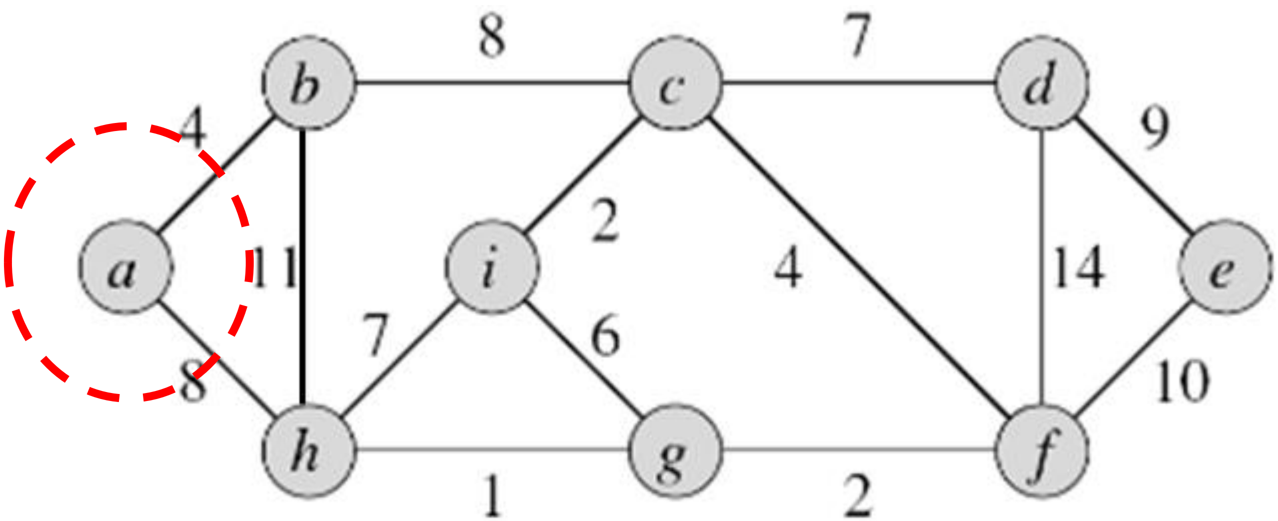
Exemplo de um Algoritmo Guloso

- S = {}
- V-S = {a, b, c, d, e, f, g, h, i}
- Solução = {}
- Aresta Leve:



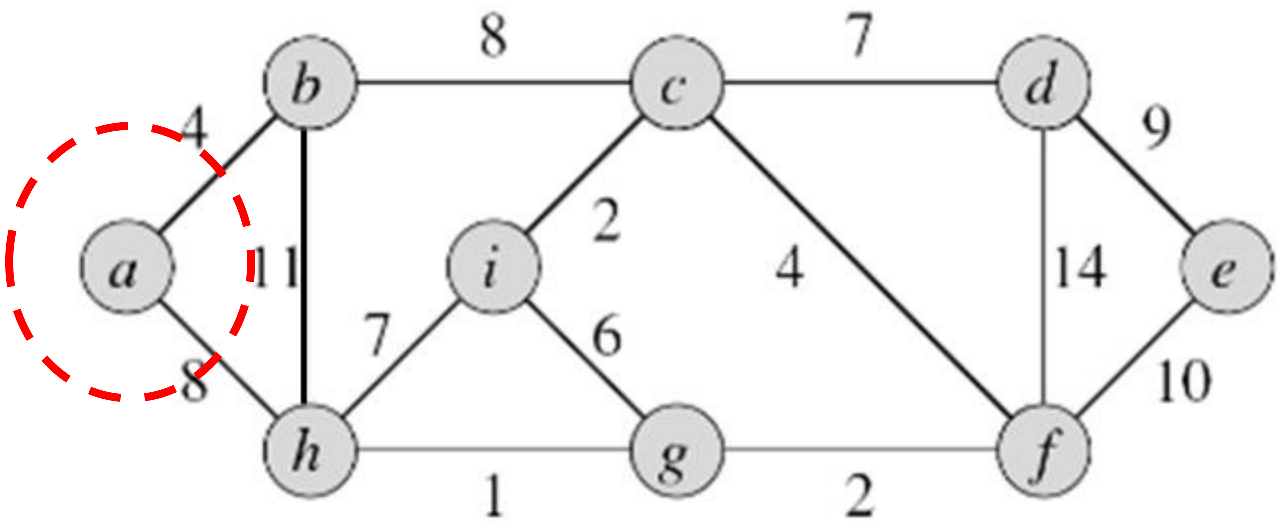
Exemplo de um Algoritmo Guloso

- S = {a}
- V-S = {b, c, d, e, f, g, h, i}
- Solução = {}
- Aresta Leve: (a,b)



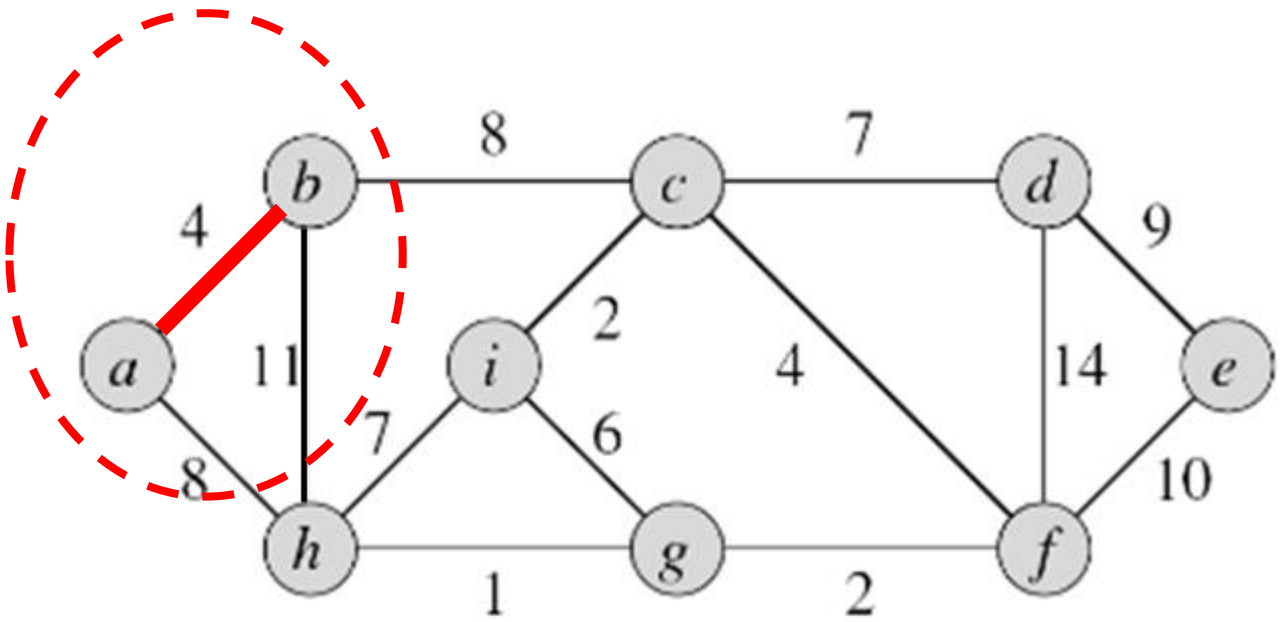
Exemplo de um Algoritmo Guloso

- S = {a, b}
- V-S = {c, d, e, f, g, h, i}
- Solução = {}
- Aresta Leve: (a,b)



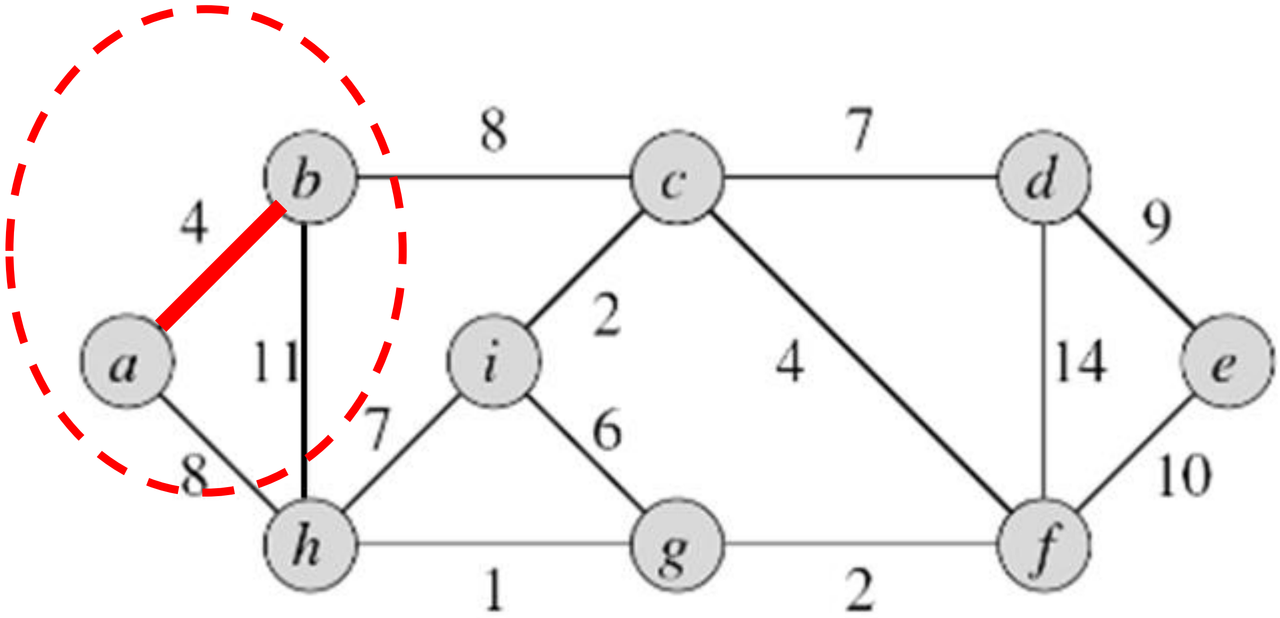
Exemplo de um Algoritmo Guloso

- S = {a, b}
- V-S = {c, d, e, f, g, h, i}
- Solução = {(a,b)}
- Aresta Leve:



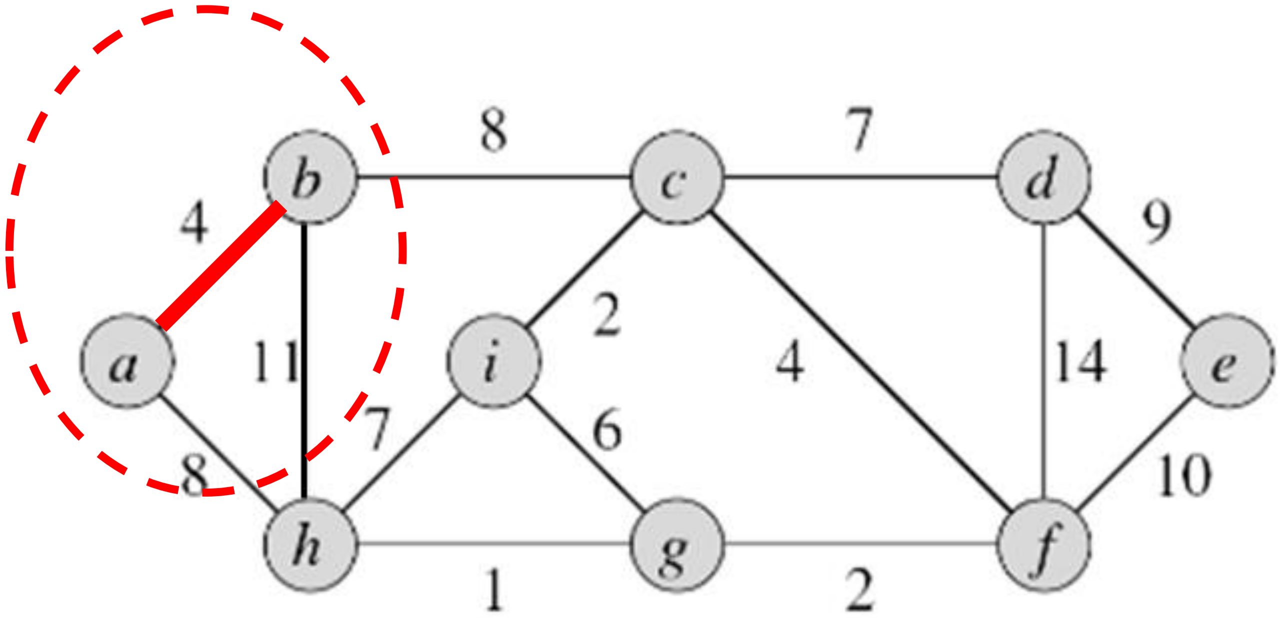
Exemplo de um Algoritmo Guloso

- S = {a, b}
- V-S = {c, d, e, f, g, h, i}
- Solução = {(a,b)}
- Aresta Leve: (b,c) e (a,h)



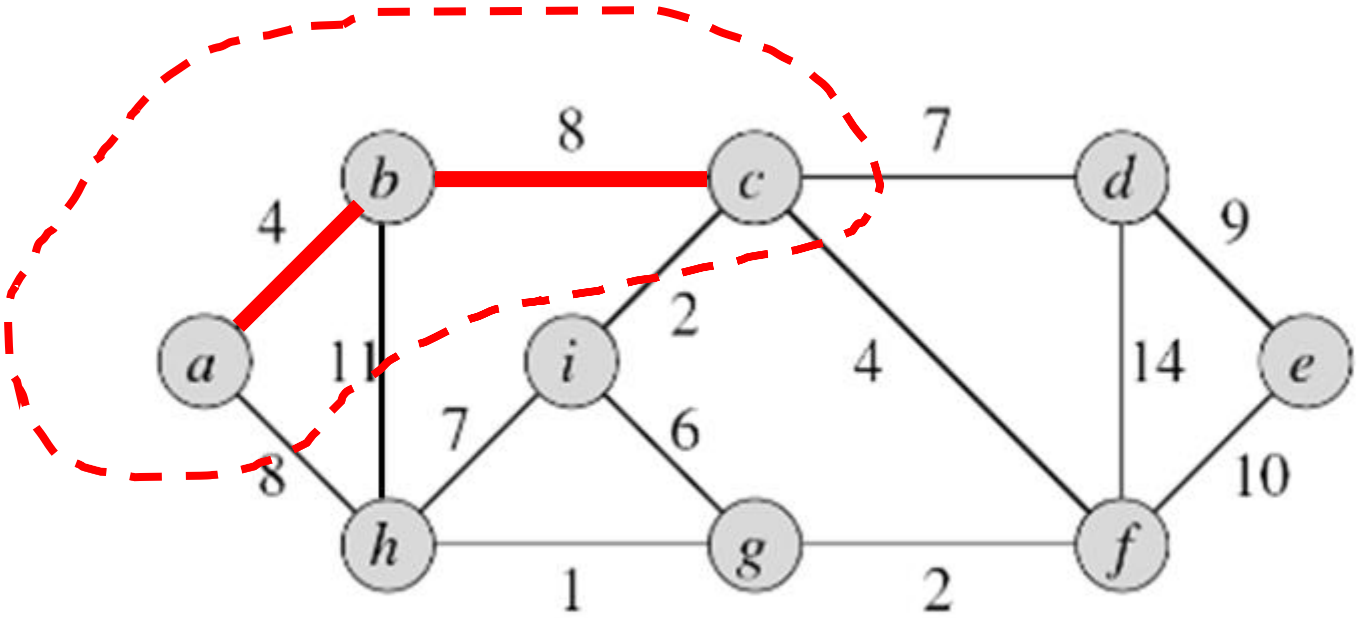
Exemplo de um Algoritmo Guloso

- S = {a, b, c}
- V-S = {d, e, f, g, h, i}
- Solução = {(a,b)}
- Aresta Leve: <u>(b,c)</u> e (a,h)



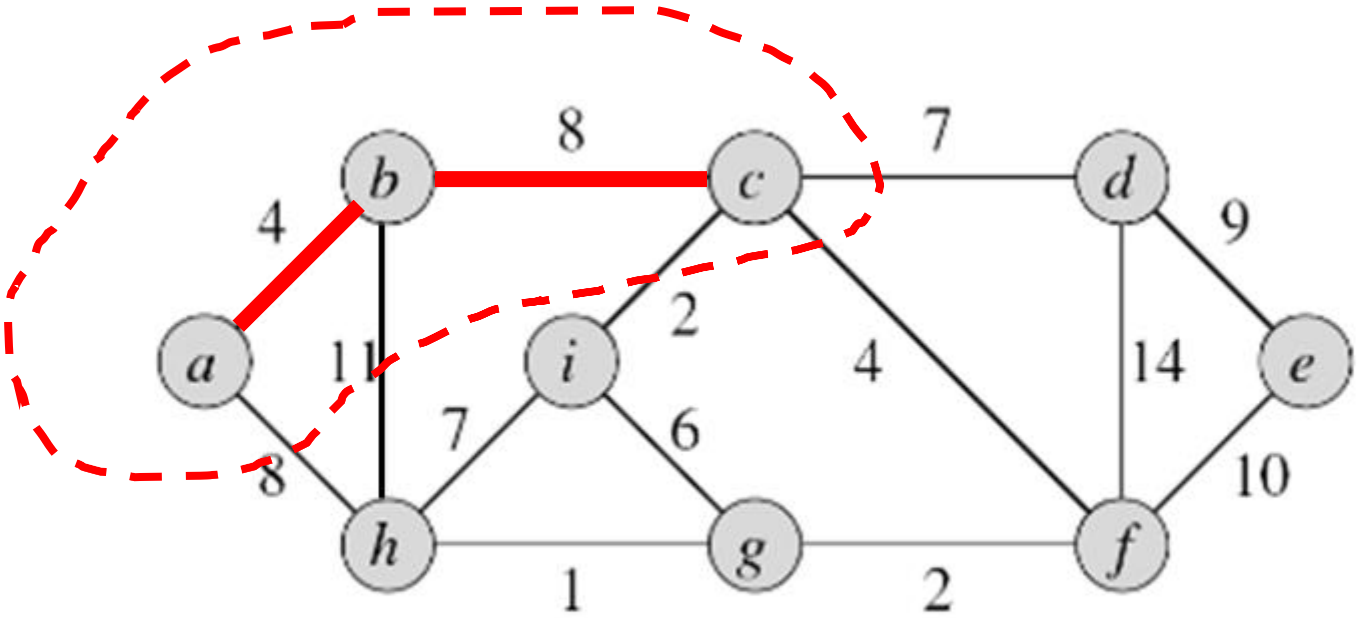
Exemplo de um Algoritmo Guloso

- S = {a, b, c}
- V-S = {d, e, f, g, h, i}
- Solução = {(a,b), (b,c)}
- Aresta Leve:



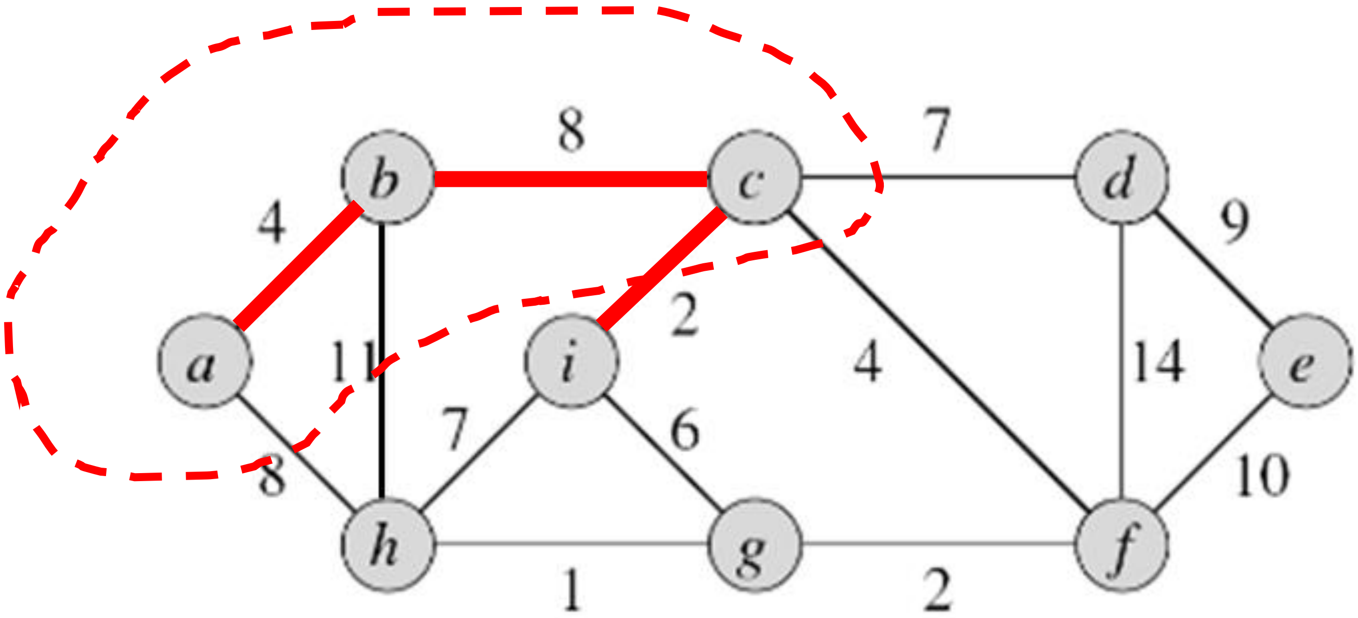
Exemplo de um Algoritmo Guloso

- S = {a, b, c}
- V-S = {d, e, f, g, h, i}
- Solução = {(a,b), (b,c)}
- Aresta Leve: (c,i)



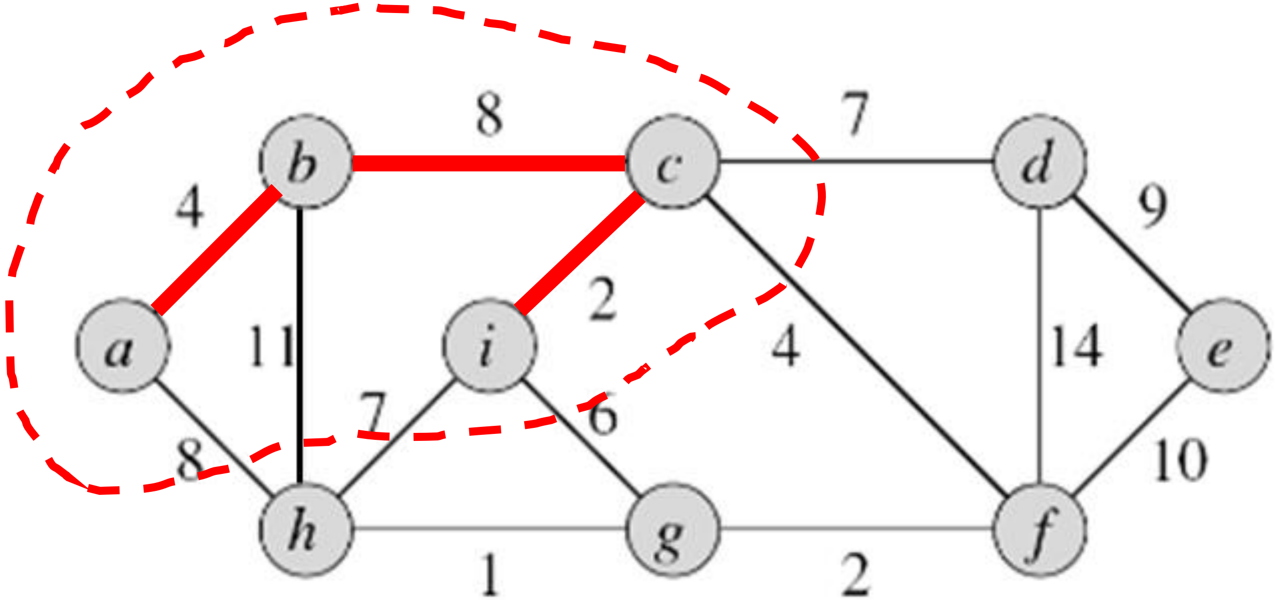
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i}
- V-S = {d, e, f, g, h}
- Solução = {(a,b), (b,c), (c,i)}
- Aresta Leve:



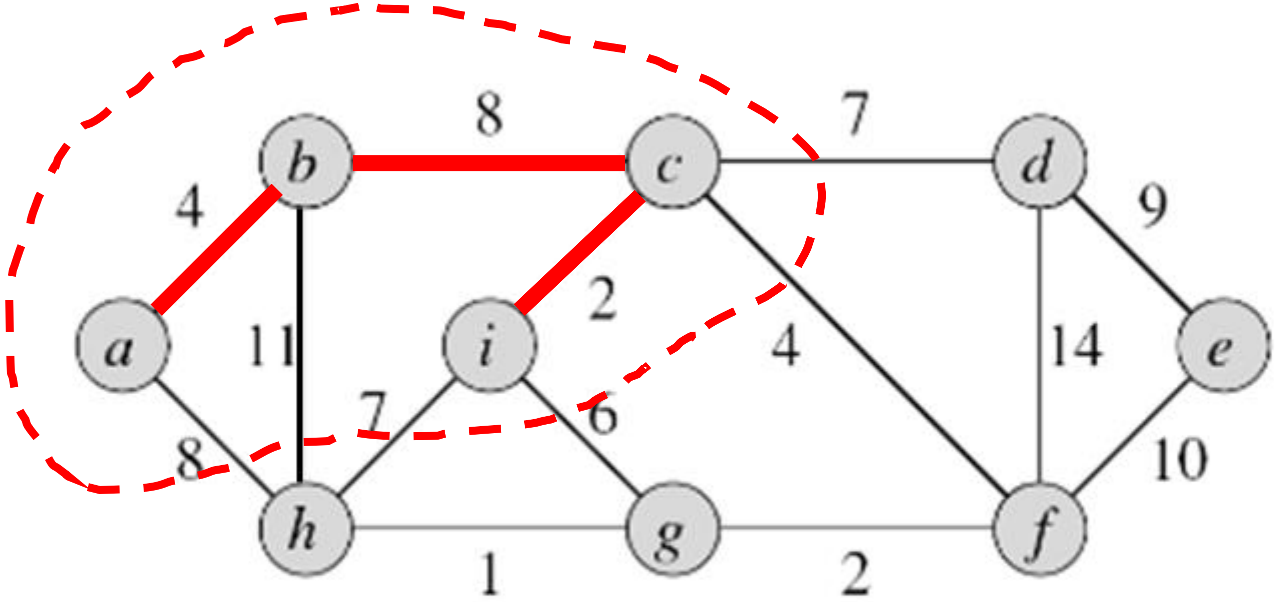
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i}
- V-S = {d, e, f, g, h}
- Solução = {(a,b), (b,c), (c,i)}
- Aresta Leve:



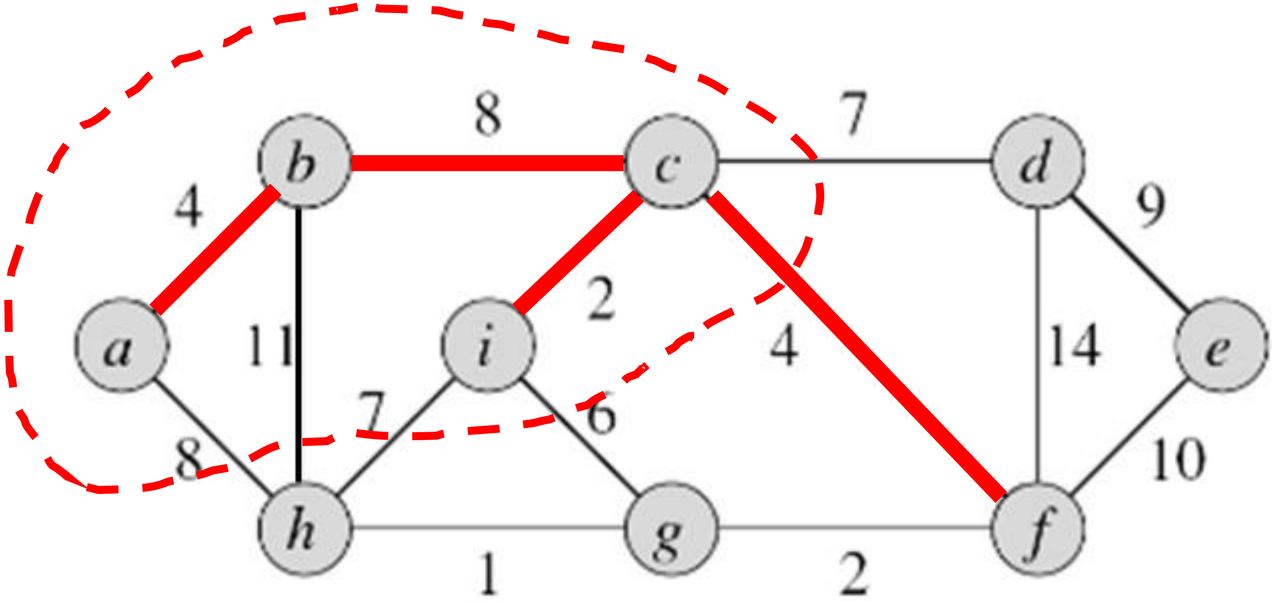
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i}
- V-S = {d, e, f, g, h}
- Solução = {(a,b), (b,c), (c,i)}
- Aresta Leve: (c,f)



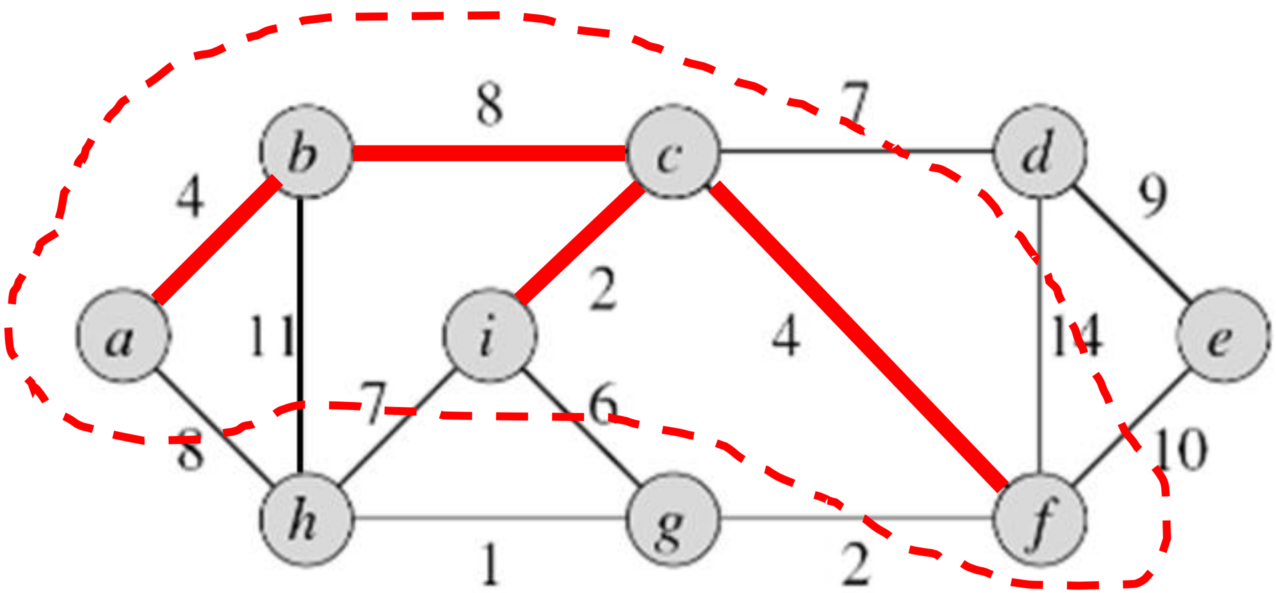
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f}
- V-S = {d, e, g, h}
- Solução = {(a,b), (b,c), (c,i), (c,f)}
- Aresta Leve:



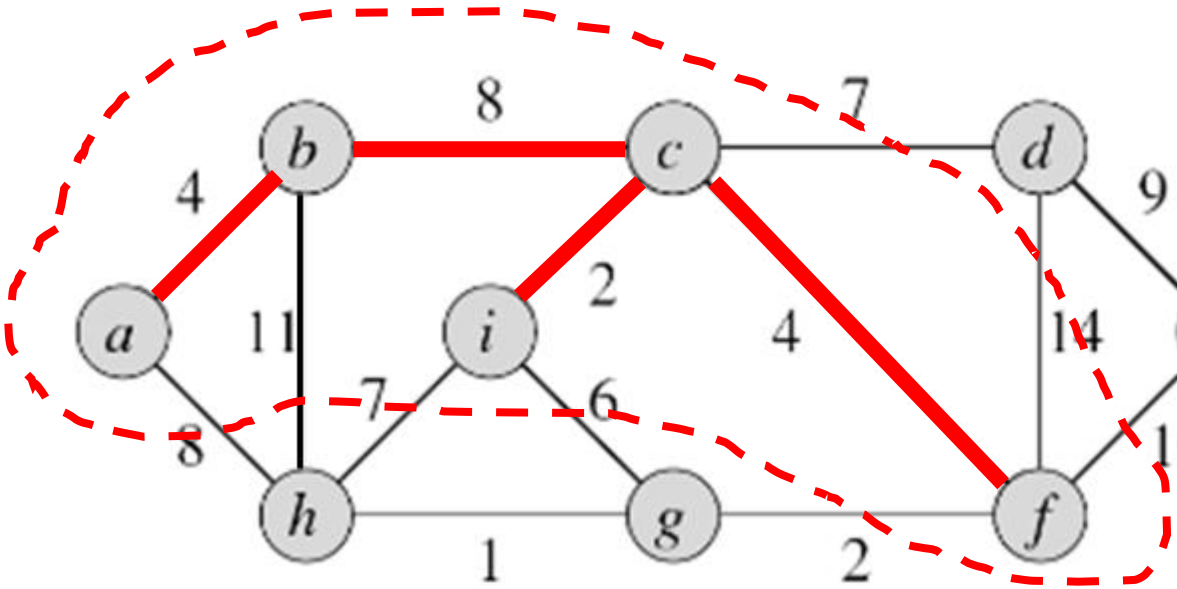
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f}
- V-S = {d, e, g, h}
- Solução = {(a,b), (b,c), (c,i), (c,f)}
- Aresta Leve:



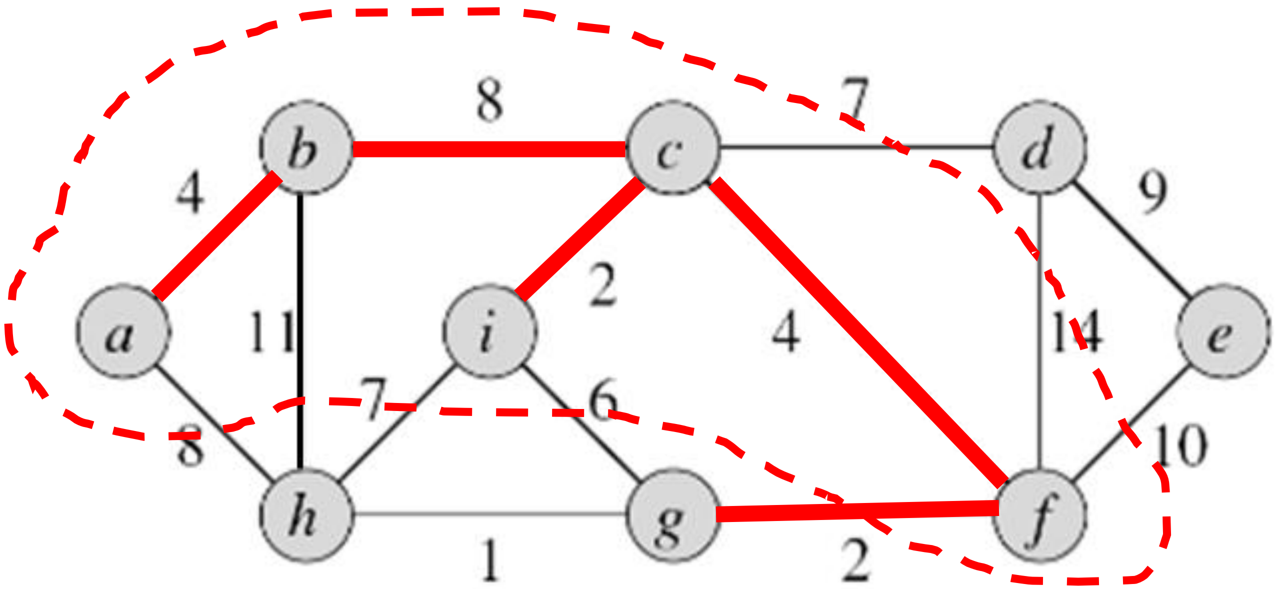
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f}
- V-S = {d, e, g, h}
- Solução = {(a,b), (b,c), (c,i), (c,f)}
- Aresta Leve: (f,g)



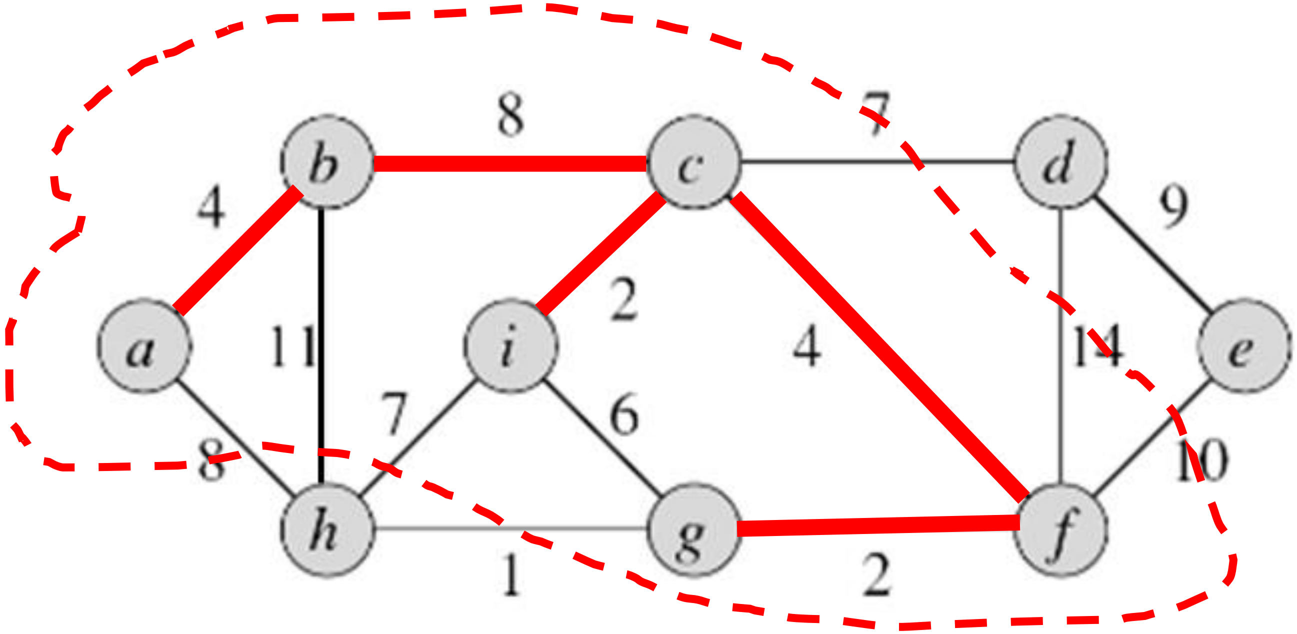
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g}
- V-S = {d, e, h}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g)}
- Aresta Leve:



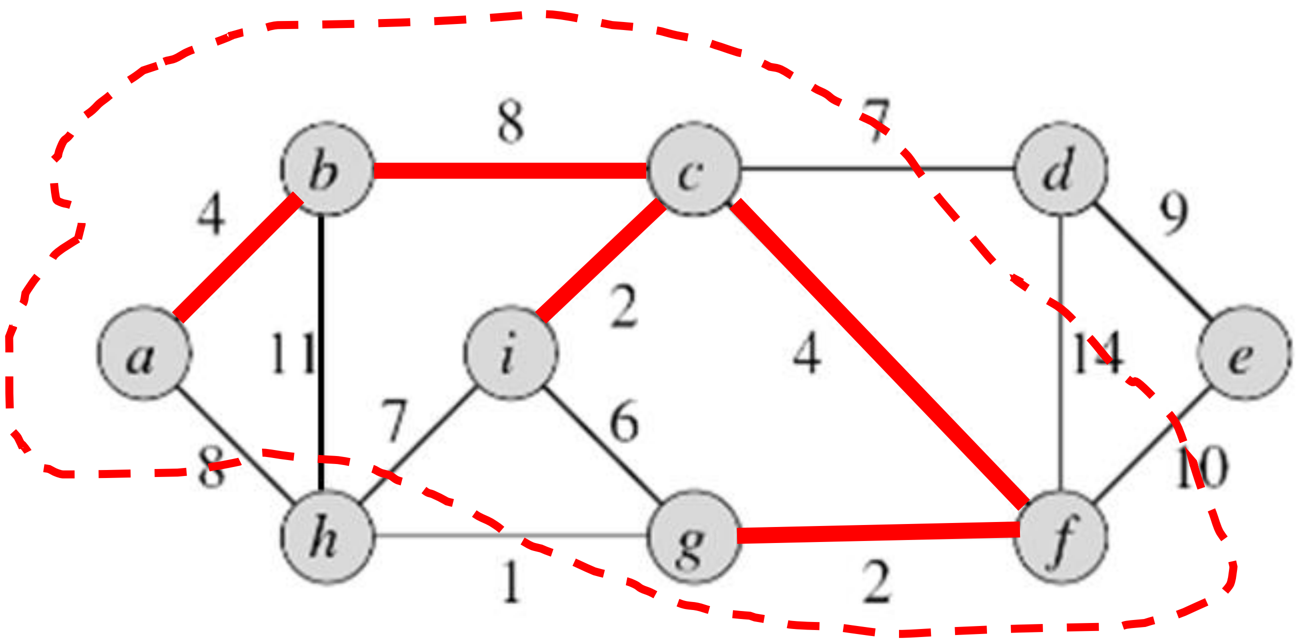
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g}
- V-S = {d, e, h}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g)}
- Aresta Leve:



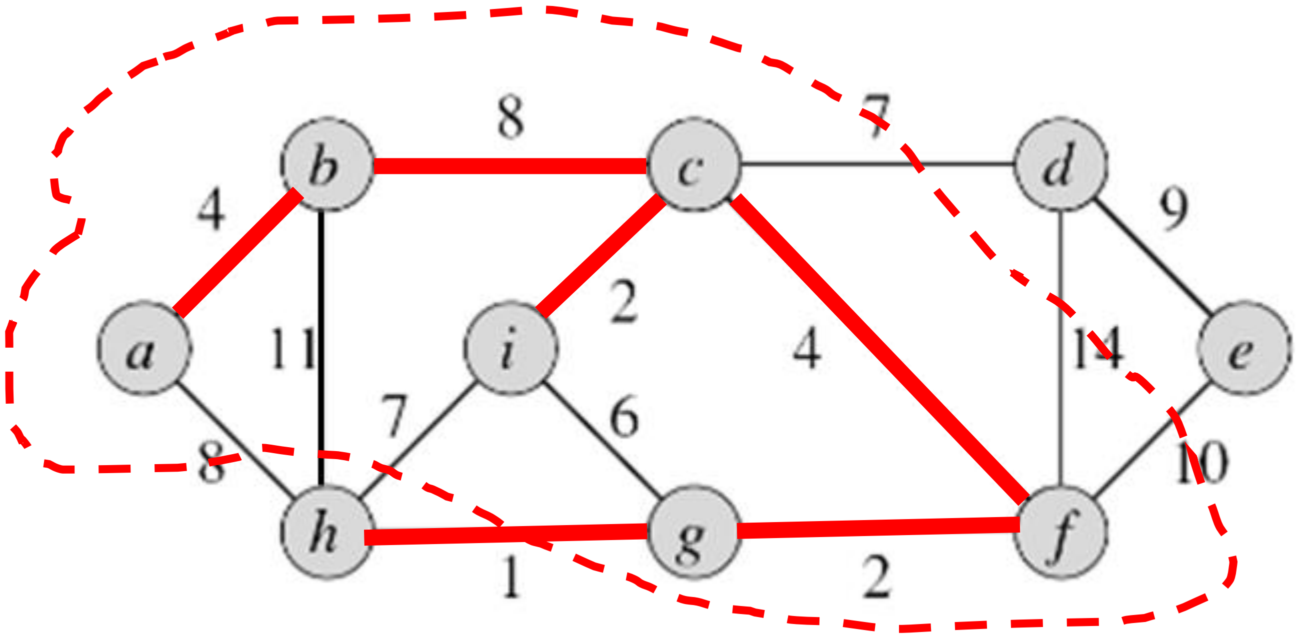
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g}
- V-S = {d, e, h}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g)}
- Aresta Leve: (g,h)



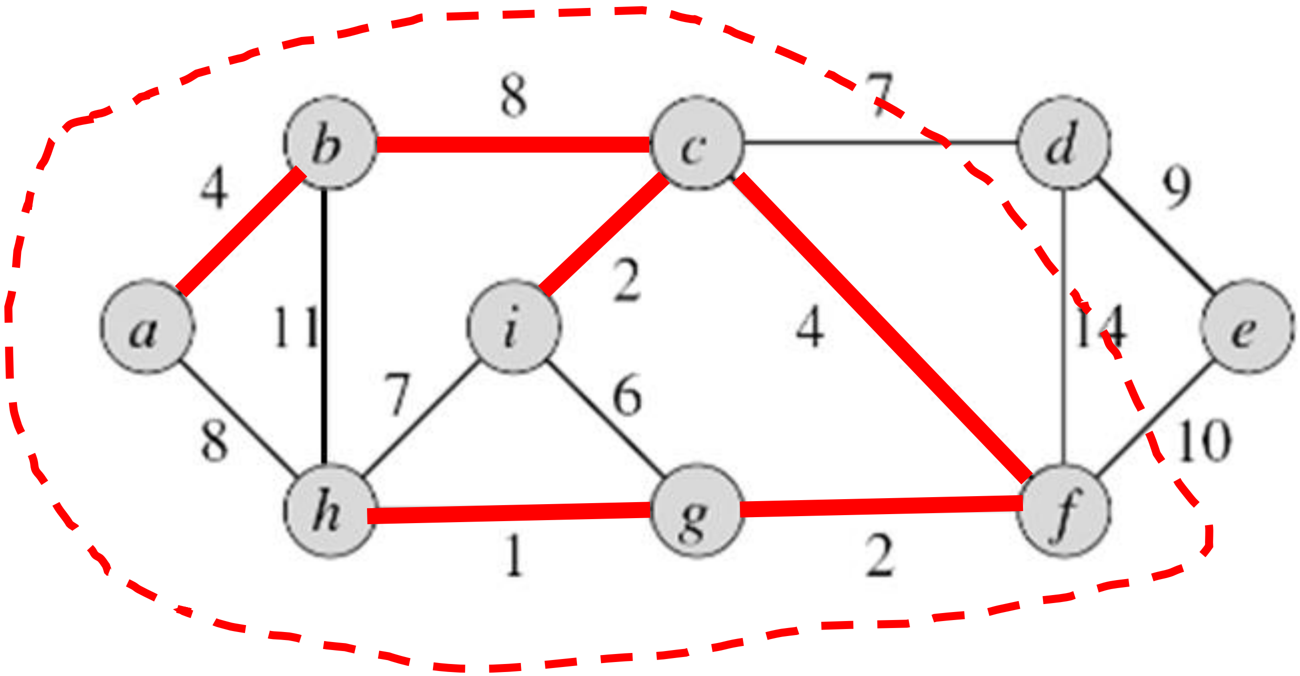
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h}
- V-S = {d, e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h)}
- Aresta Leve:



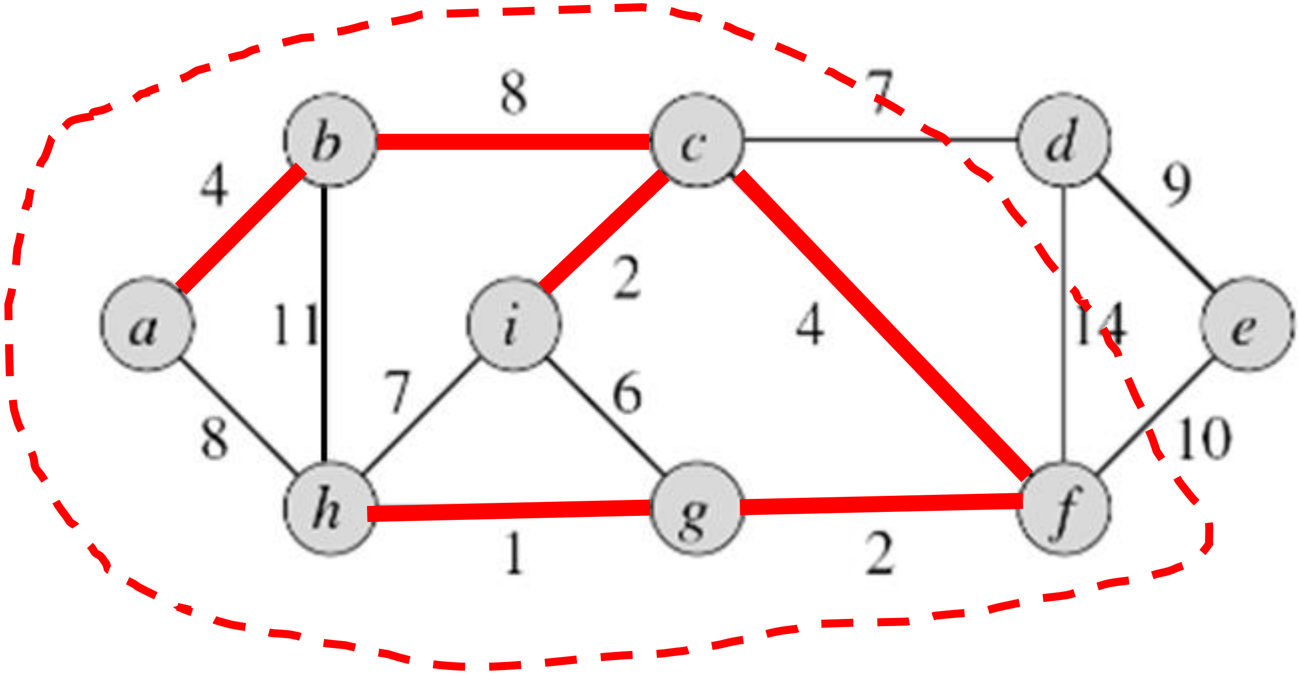
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h}
- V-S = {d, e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h)}
- Aresta Leve:



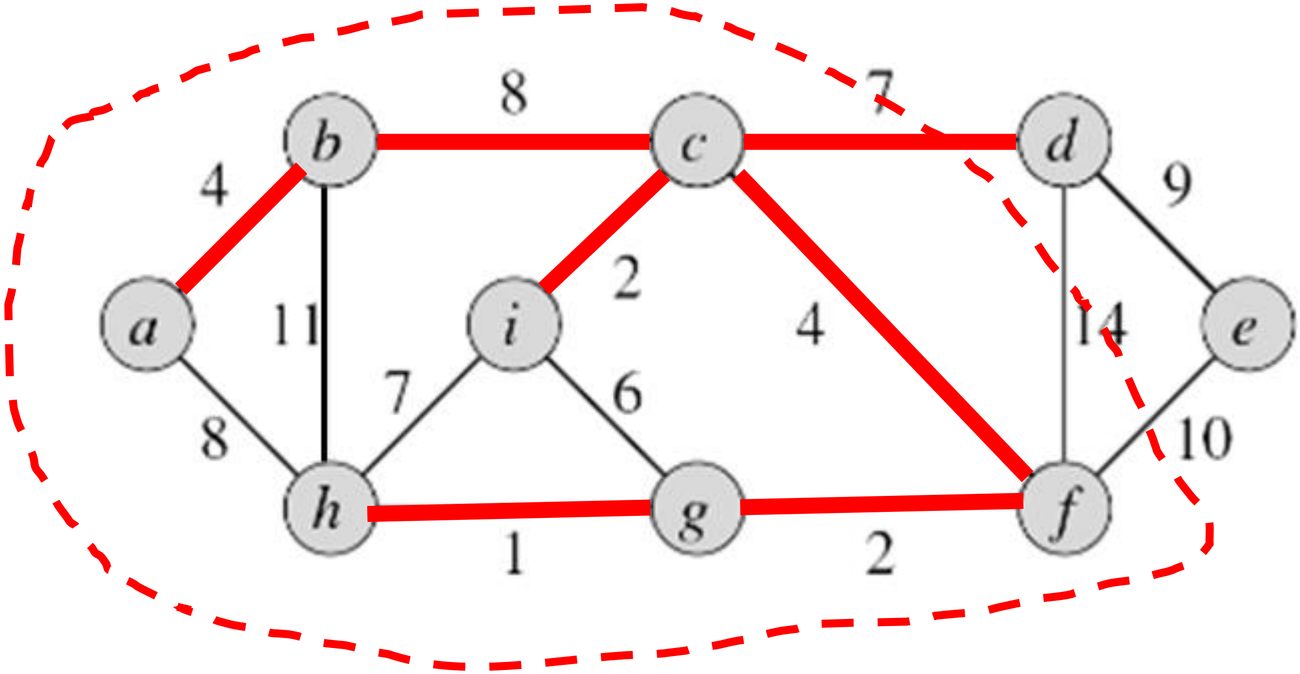
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h}
- V-S = {d, e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h)}
- Aresta Leve: (c,d)



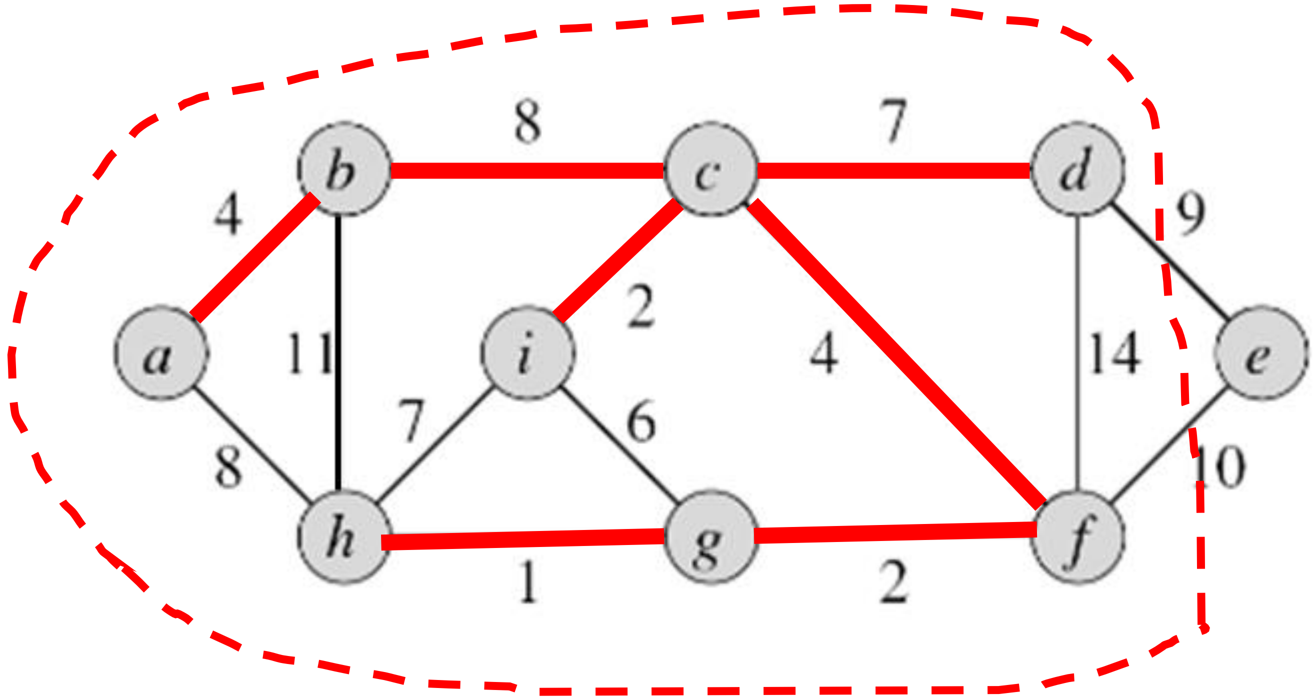
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h, d}
- V-S = {e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h), (c,d)}
- Aresta Leve:



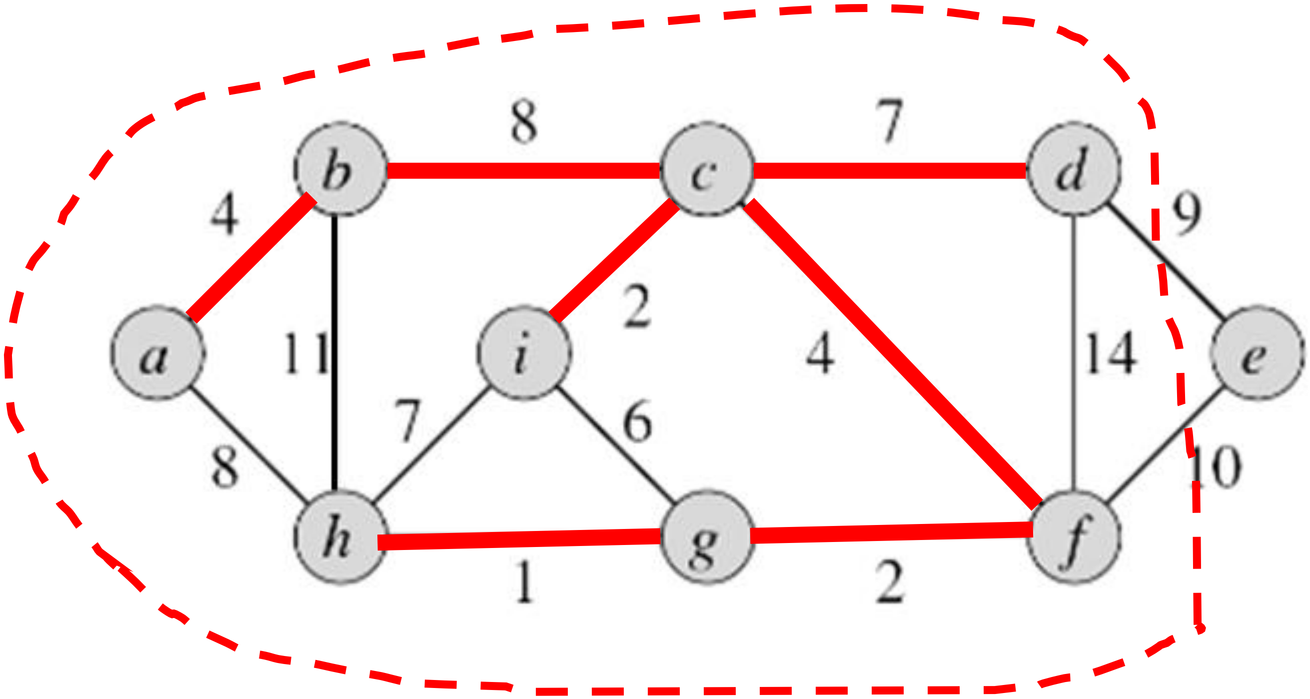
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h, d}
- V-S = {e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h), (c,d)}
- Aresta Leve:



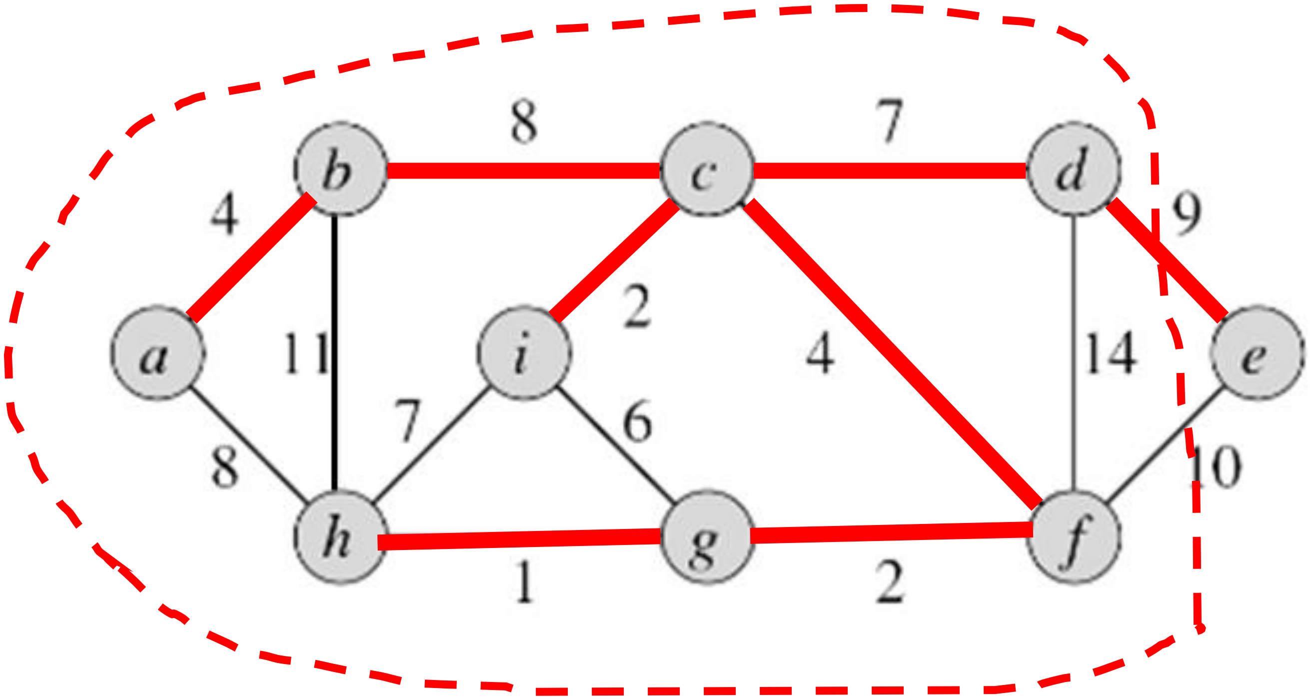
Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h, d}
- V-S = {e}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h), (c,d)}
- Aresta Leve: (d,e)



Exemplo de um Algoritmo Guloso

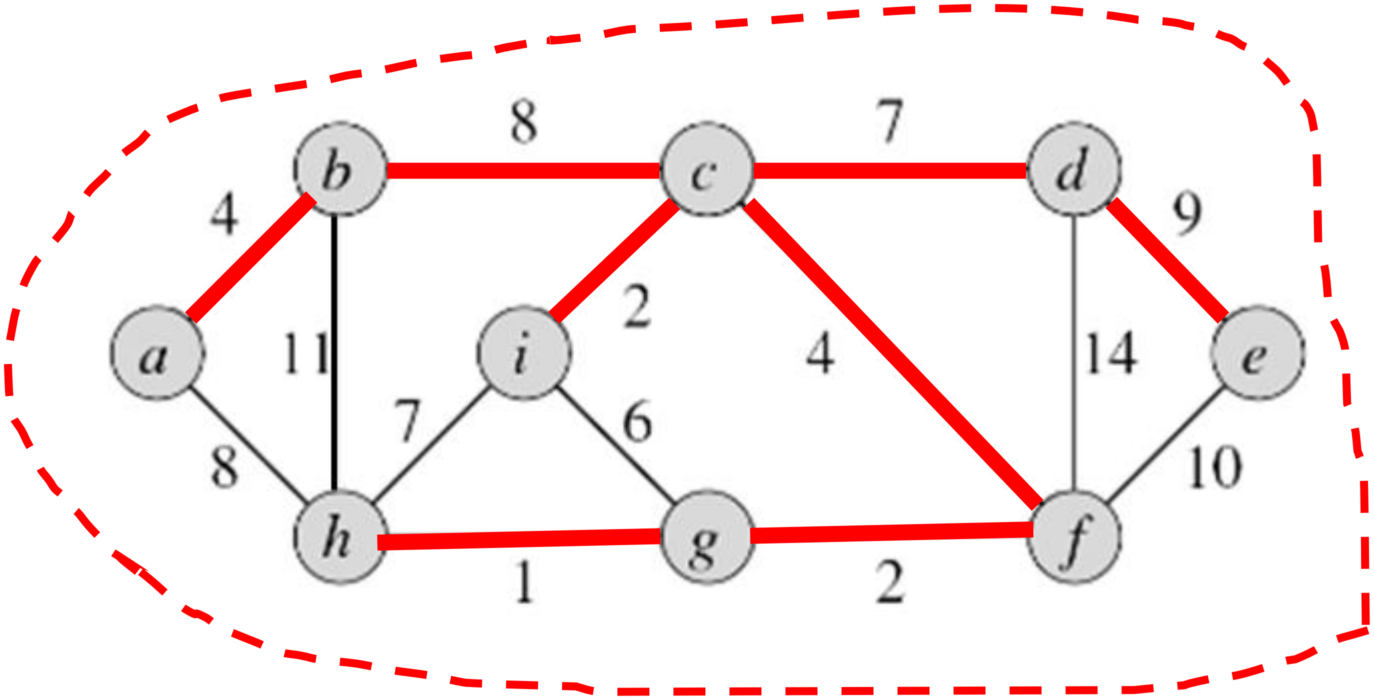
- S = {a, b, c, i, f, g, h, d, e}
- V-S = {}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h), (c,d), (d,e)}
- Aresta Leve:



unesp
DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Exemplo de um Algoritmo Guloso

- S = {a, b, c, i, f, g, h, d, e}
- V-S = {}
- Solução = {(a,b), (b,c), (c,i), (c,f), (f,g), (g,h), (c,d), (d,e)}
- Aresta Leve:



Exemplo de um Algoritmo Guloso

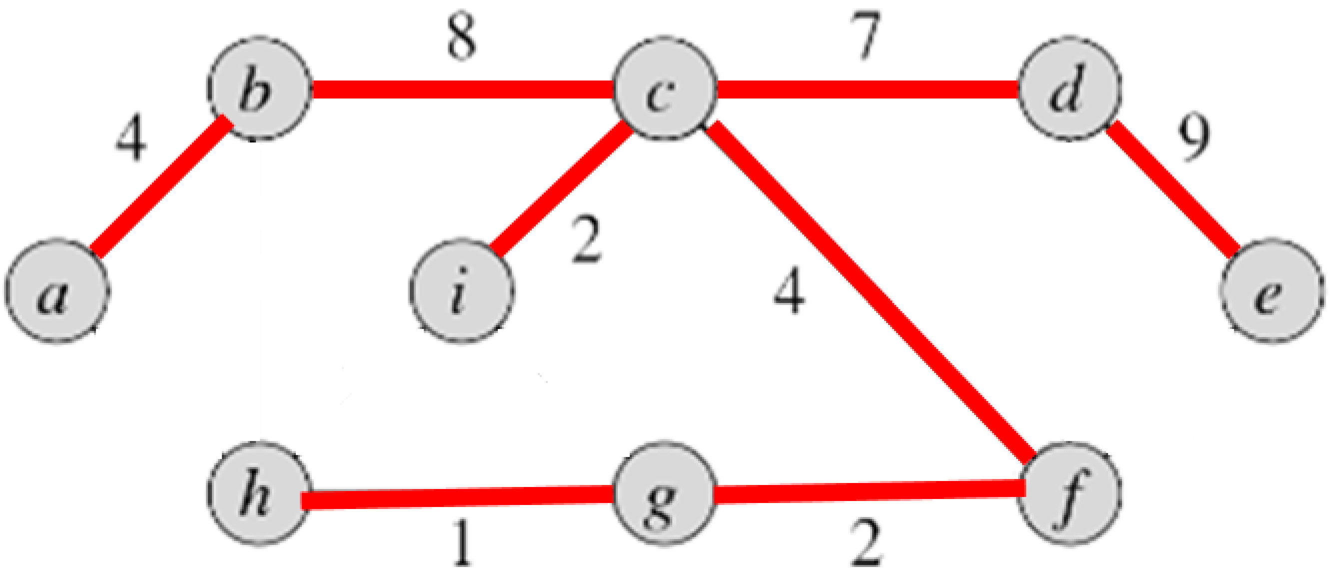
Algoritmo PRIM

1. Selecionar um vértice inicial para o conjunto S
   1. Assim, um corte inicial é definido
2. Encontrar a aresta leve (i,j) que cruza o corte e adicionar como solução
3. Atualizar o conjunto S adicionando o vértice ‘i’ ou ‘j’, isto é, aquele que não pertencia ao conjunto S
   1. Assim, o corte foi atualizado para que as arestas do conjunto solução não cruzem o corte
4. Se V-S diferente de vazio, repetir a partir do Passo 2

DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Exemplo de um Algoritmo Guloso

- Árvore Geradora Mínima
  - Custo 37



Bibliografia

- LEVITIN, A. Introduction to The Design & Analysis of Algorithms, Addison Wesley, 2003, 497p.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; (2002). Algoritmos –Teoria e Prática. Tradução da 2ª edição americana. Rio de Janeiro. Editora Campus
- TAMASSIA, ROBERTO; GOODRICH, MICHAEL T. (2004). Projeto de Algoritmos -Fundamentos, Análise e Exemplos da Internet
- ZIVIANI, N. (2007). Projeto e Algoritmos com implementações em Java e C++. São Paulo. Editora Thomson

unesp

DEPARTAMENTO DE MATEMÁTICA E COMPUTAÇÃO

Bibliografia

BIBLIOGRAFIA BÁSICA
1. CORMEN, T. H.; LEISERSON C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos**: teoria e prática, 2ª ed. Rio de Janeiro: Editora Campus, 2002. 916p. 2. SEDGEWICK, R. **Algorithms in C**: graph algorithms, 3ª ed. Part 5, Addison Wesley, 2002. 482p. 3. NETTO, P. O. B. **Grafos**: teoria, modelos, algoritmos. 2ª ed. São Paulo: Editora Edgard Blücher Ltda, 2001. 304p. 4. SZWARCFITER, J. L. **Grafos e Algoritmos Computacionais**. Rio de Janeiro: Editora Campus, 1984. 216p. 5. ZIVIANI, N. **Projeto de algoritmos**: com implementações em PASCAL e C. 2ª ed. São Paulo: Pioneira Thomson Learning Ltda, 2004. 552p.

BIBLIOGRAFIA COMPLEMENTAR
1. DROZDEK, A. **Estrutura de Dados e Algoritmos em C++**. São Paulo: Pioneira Thomson Learning Ltda, 2002. 579p. 2. GOODRICH, M. T.; TAMASSIA, R. **Estruturas de Dados e Algoritmos em JAVA**. 2ª ed. Bookman Companhia Editora, 2002. 584p. 3. SCHEINERMAN, E. R. **Matemática discreta**: uma introdução. São Paulo: Pioneira Thomson Learning, 2003. 532 p. 4. WILSON, R. J.; WATKINS, J. J. **Graphs**: an introductory approach. New York: John Wiley & Sons, Inc, 1990. 340p.