AULA 16

Prof. Mathias

Método guloso – Algoritmo de Kruskal

Análise de Algoritmos

Agenda

- Aula anterior
- Introdução
- Método guloso
- Algoritmo de Kruskal
- Exercícios
- Próxima aula

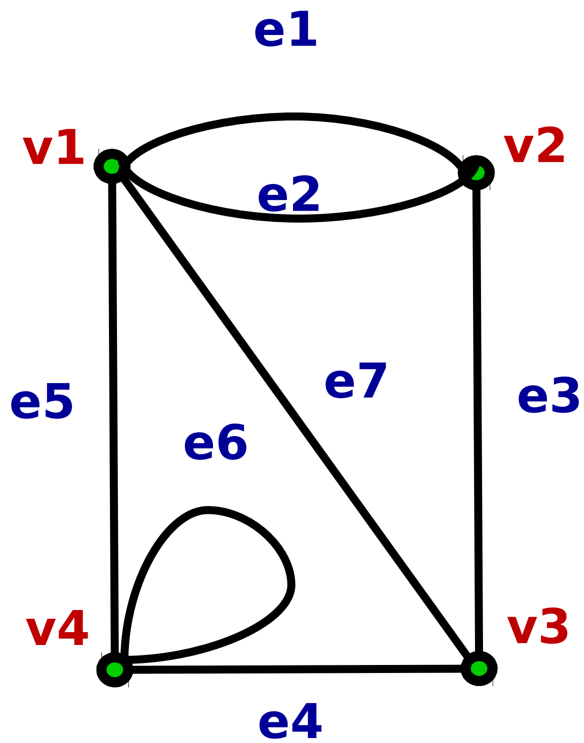
Aula Anterior

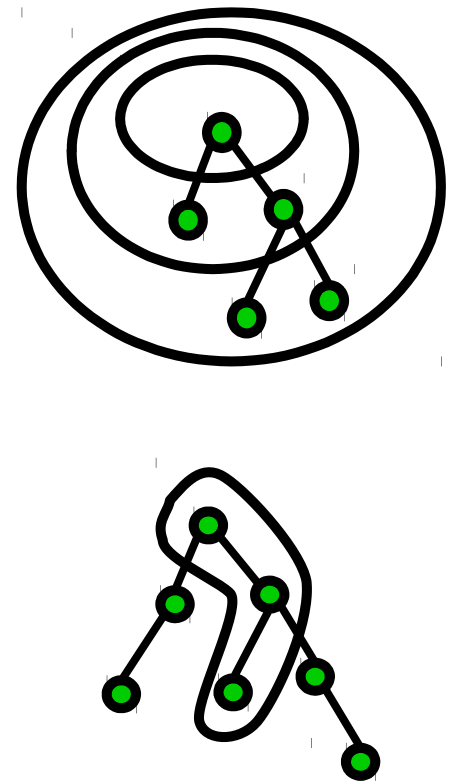
- Busca em grafos
  - Largura
  - Profundidade

Agenda

- Aula anterior
- Introdução

Introdução





Introdução

- Método guloso:
  - É útil principalmente para resolver problemas de otimização combinatória, cuja soluções possam ser alcançadas por sequências de decisões.
  - Toscani & Veloso(2005)

Agenda

- Aula anterior
- Introdução
- Método guloso

Método Guloso

- Podemos aplicar em:
  - Árvore geradora
  - Intercalação sucessiva ótima de listas
  - Caminhos de custo mínimo de grafos orientados
  - Escalonamento de tarefas

Método Guloso

- Idéias básicas:
  - Construir por etapas uma resposta ótima
  - Em cada passo, após selecionar um elemento de entrada (o melhor), decide-se se ele é viável – vindo a fazer parte da resposta ou não.

Método Guloso

- Idéias básicas:
  - Construir por etapas uma resposta ótima
  - Em cada passo, após selecionar um elemento de entrada (o melhor), decide-se se ele é viável – vindo a fazer parte da resposta ou não.

Método Guloso

- Idéias básicas:
  - Após a sequência de decisões a resposta é
  - Nessa sequência de decisões nenhum elemento é examinado mais de uma vez, ou fará parte da saída ou será descartado.

Método Guloso

- Exemplo:
  - Uma árvore geradora (ou de espalhamento) de um grafo (não orientado) G é um subgrafo acíclico que contem todos os nodos (vértices) do grafo.Uma árvore geradora (ou de espalhamento) de um grafo (não orientado) G é um subgrafo acíclico que contem todos os nodos (vértices) do grafo.

Método Guloso

- Vejamos:
  - Se o grafo não é conexo cada componente terá uma árvore geradora
  - Se as arestas são valoradas, uma questão interessante é encontrar uma árvore geradora de custo mínimo.

Método Guloso

- Vejamos:
  - As arestas serão consideradas em ordem não descrecentes de seus custos
  - A cada passo seleciona-se uma aresta de custo mínimo dentre as arestas ainda não examinadas

Método Guloso

- Vejamos:
  - Se sua inclusão criar um ciclo ela é descartada, caso contrário ela é incluida, talvez ligando duas arvores da floresta.

Agenda

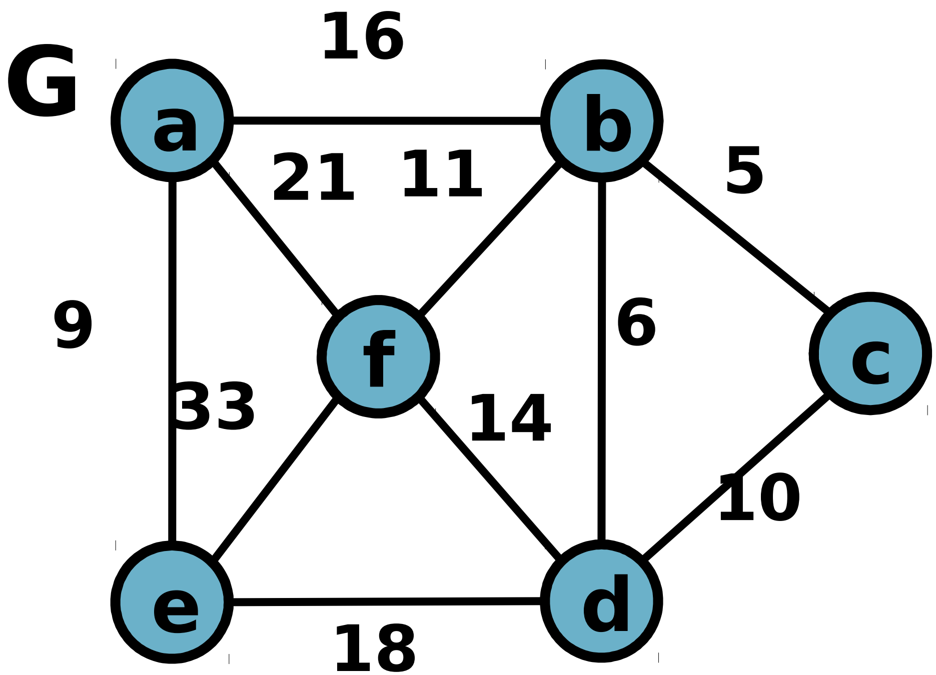
- Aula anterior
- Introdução
- Método guloso
- Algoritmo de Kruskal

Algoritmo de Kruskal

- Árvore geradora de custo mínimo:
  - Dado um grafo G(V,E), com pesos nas arestas, determinar um subgrafo gerador conexo de custo mínimo, ou seja, uma arvore geradora de custo mínimo.

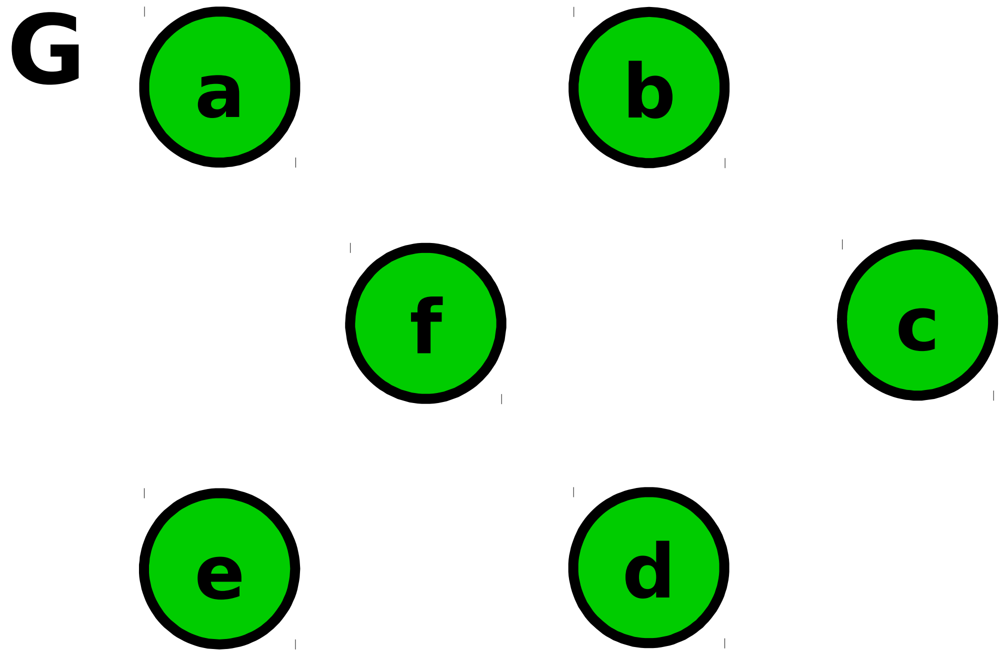
Algoritmo de Kruskal

- Árvore geradora de custo mínimo:



Algoritmo de Kruskal

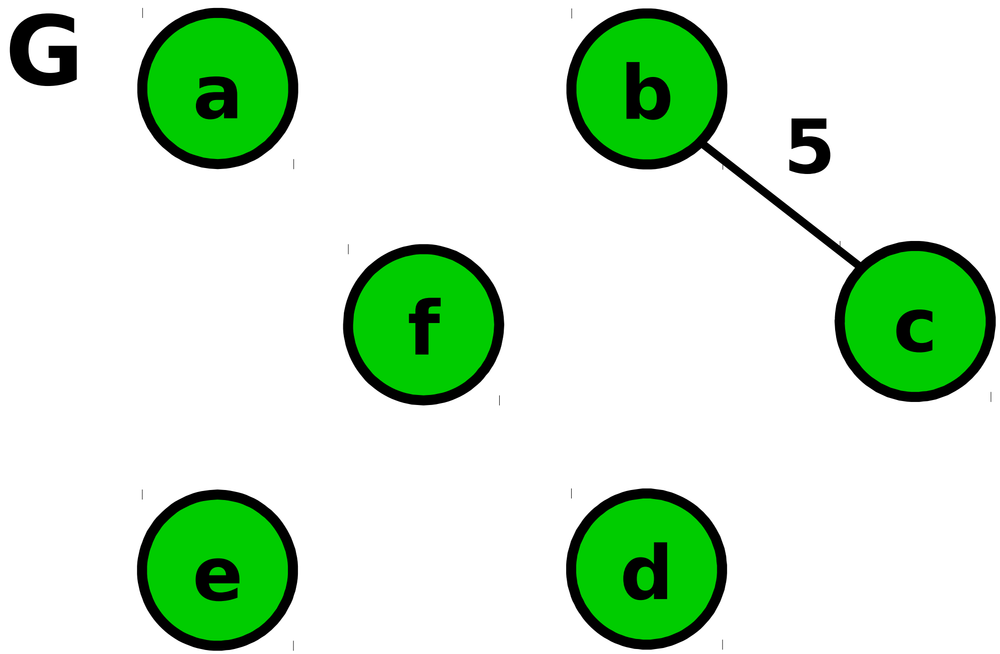
- Árvore geradora de custo mínimo:



C = {(a), (b), (c), (d), (e), (f) }

Algoritmo de Kruskal

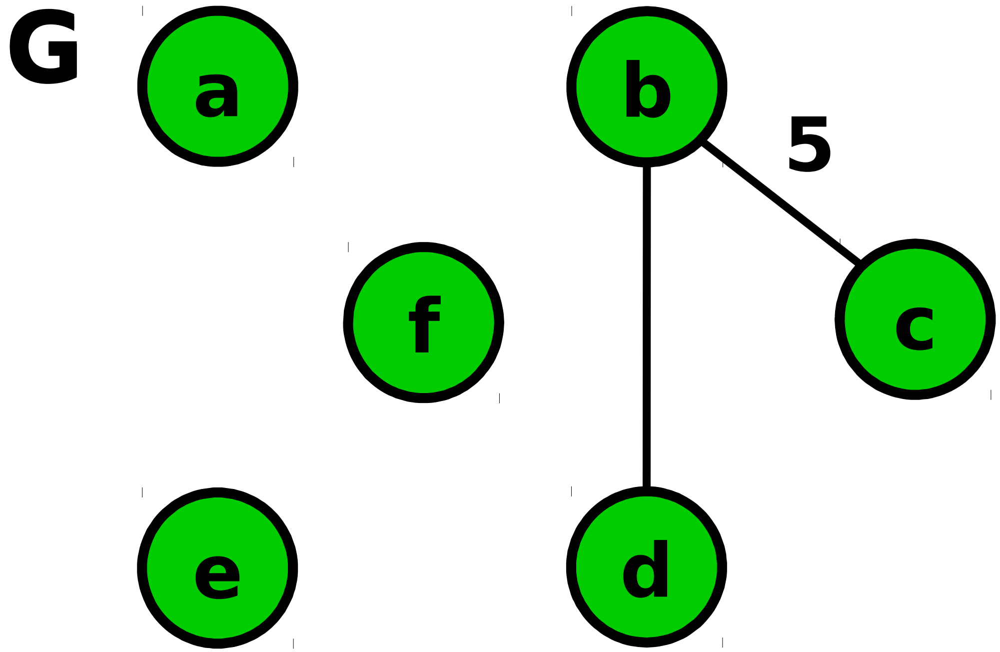
- Árvore geradora de custo mínimo:



C = {(a), (b, c), (d), (e), (f) }

Algoritmo de Kruskal

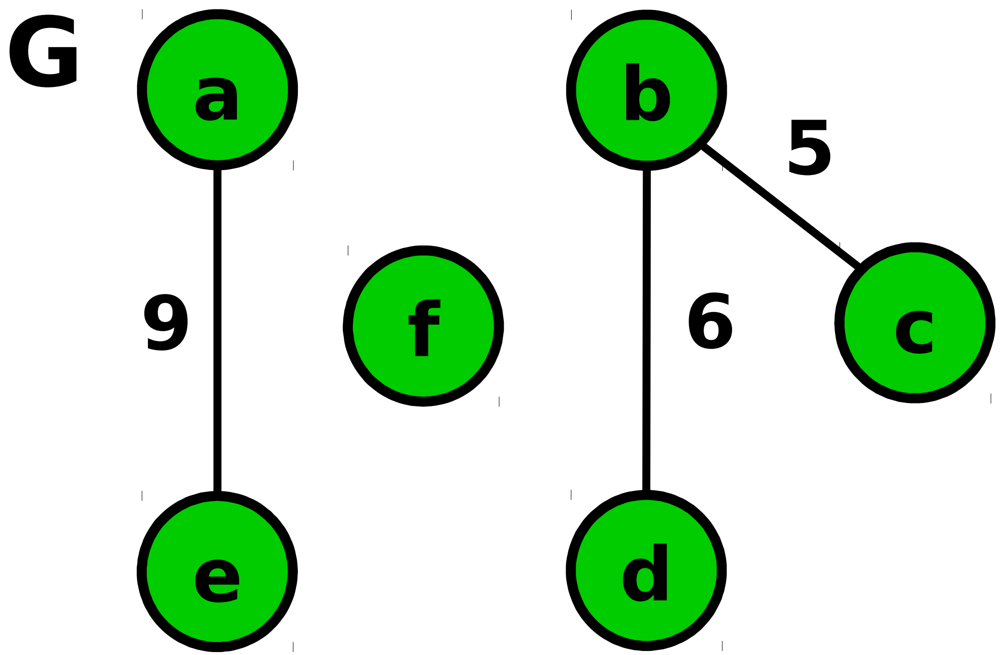
- Árvore geradora de custo mínimo:



C = {(a), (b, c, d), (e), (f) }

Algoritmo de Kruskal

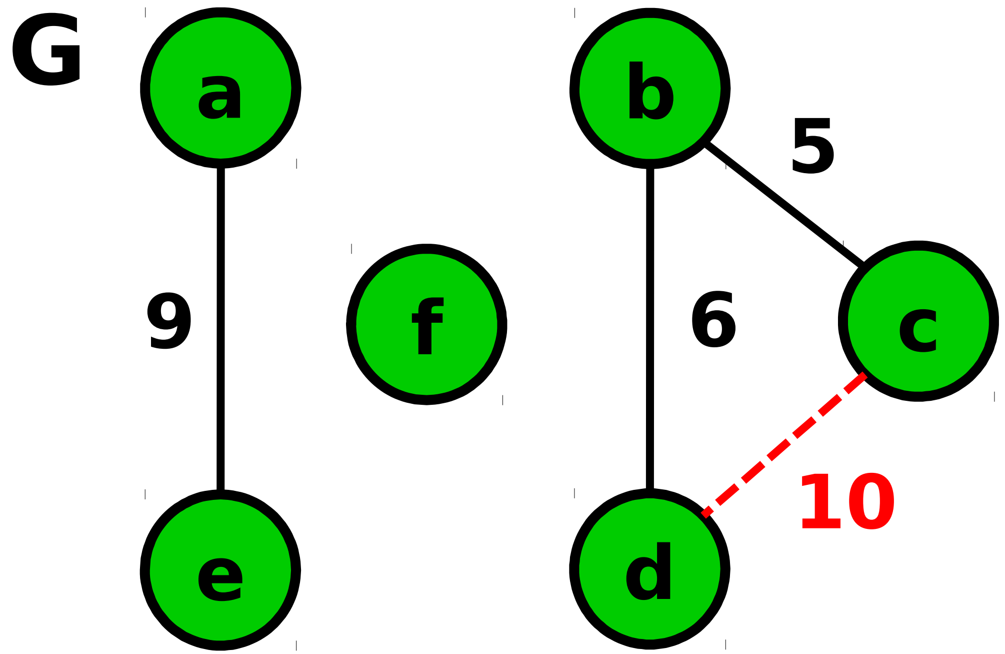
- Árvore geradora de custo mínimo:



C = {(a, e), (b, c, d), (f) }

Algoritmo de Kruskal

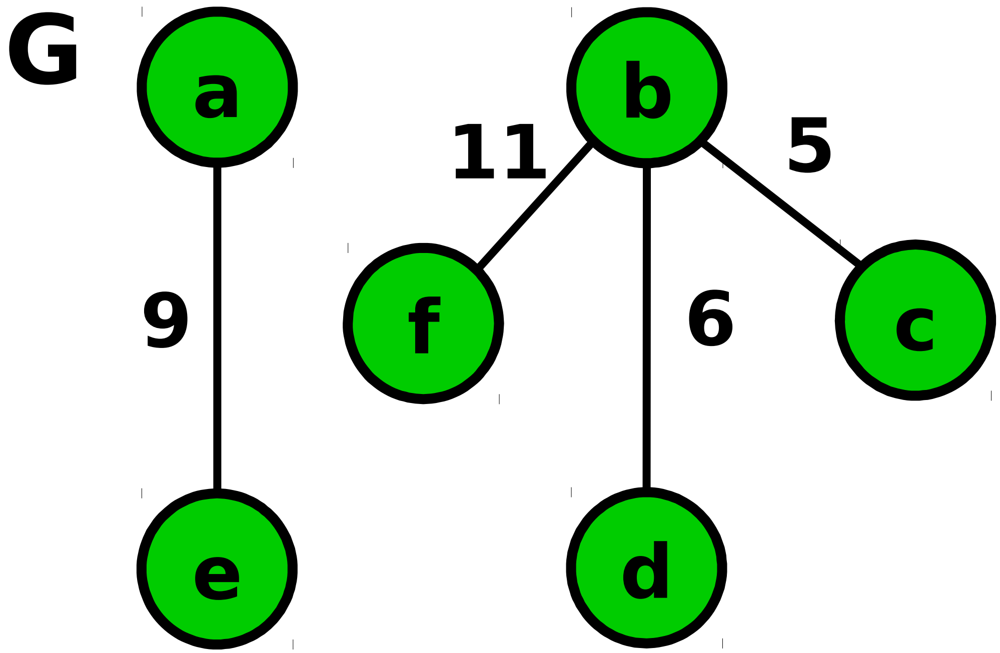
- Árvore geradora de custo mínimo:



C = {(a, e), (b, c, d), (f) }

Algoritmo de Kruskal

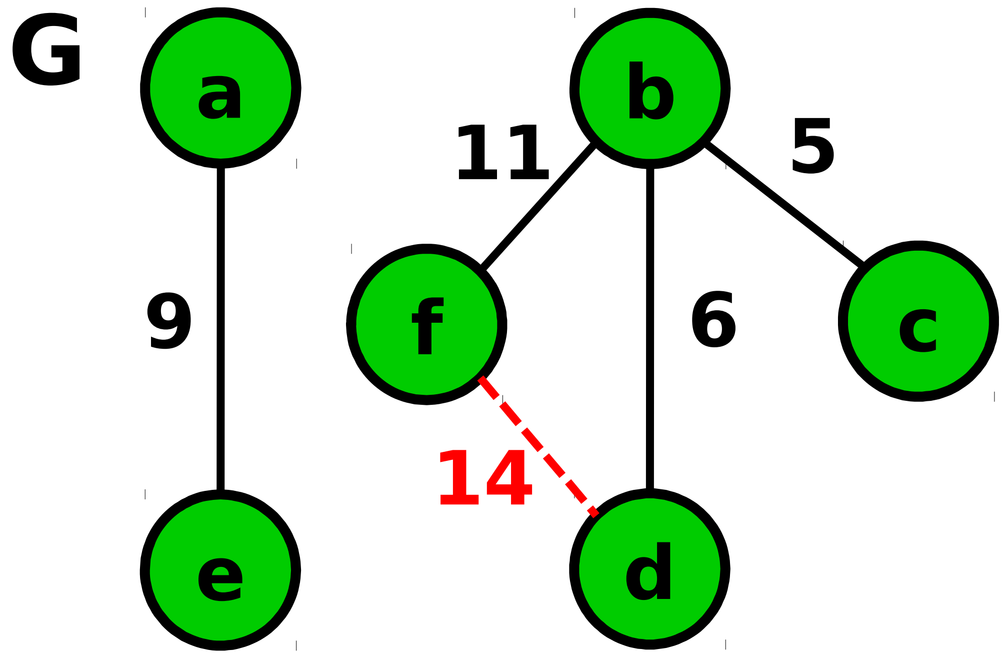
- Árvore geradora de custo mínimo:



C = {(a, e), (b, c, d, f) }

Algoritmo de Kruskal

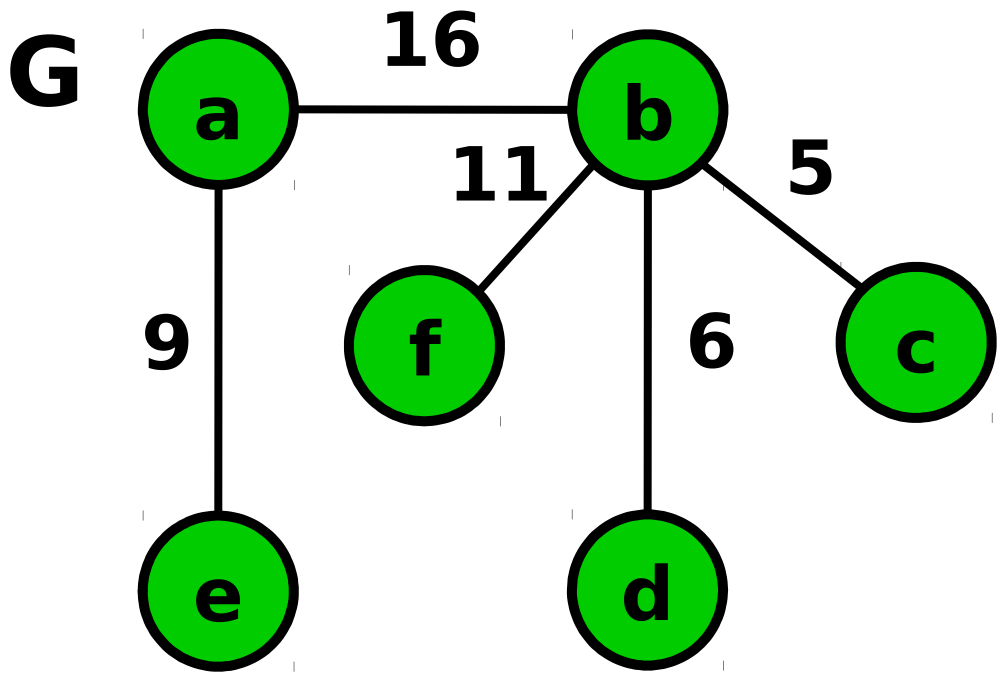
- Árvore geradora de custo mínimo:



C = {(a, e), (b, c, d, f) }

Algoritmo de Kruskal

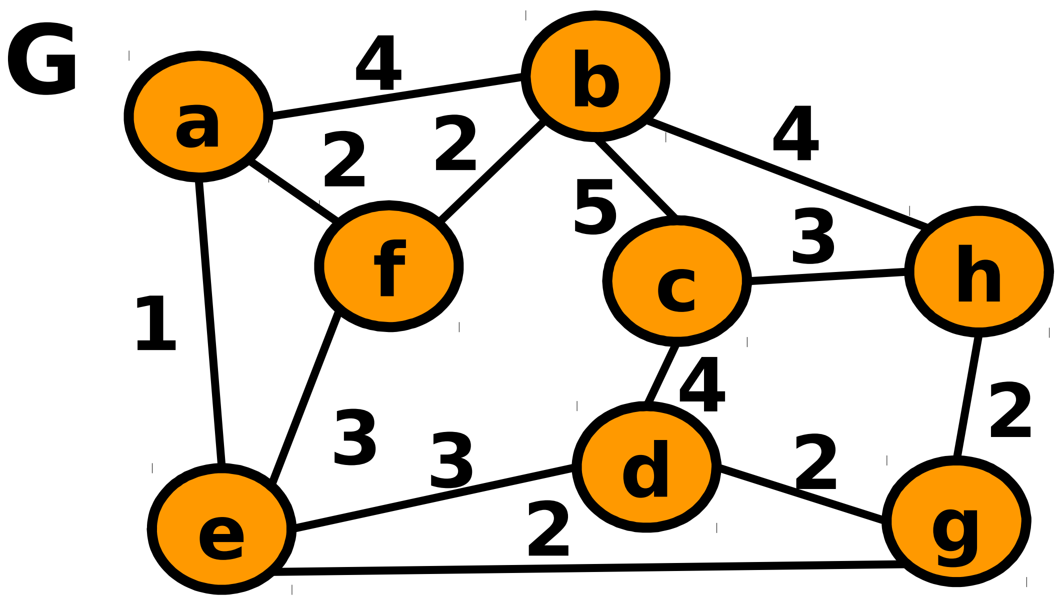
- Árvore geradora de custo mínimo:



C = {(a, e, b, c, d, f) }

Algoritmo de Kruskal

- Como seria a árvore geradora de custo mínimo deste grafo?



C = {(a, b, c, d, e, f, g, h) }

Algoritmo de Kruskal

- Implementação

1. Dado um grafo ***G = (V, E)*** considere o grafo

T = (V(E),O)

2. ***S <- E(G)***

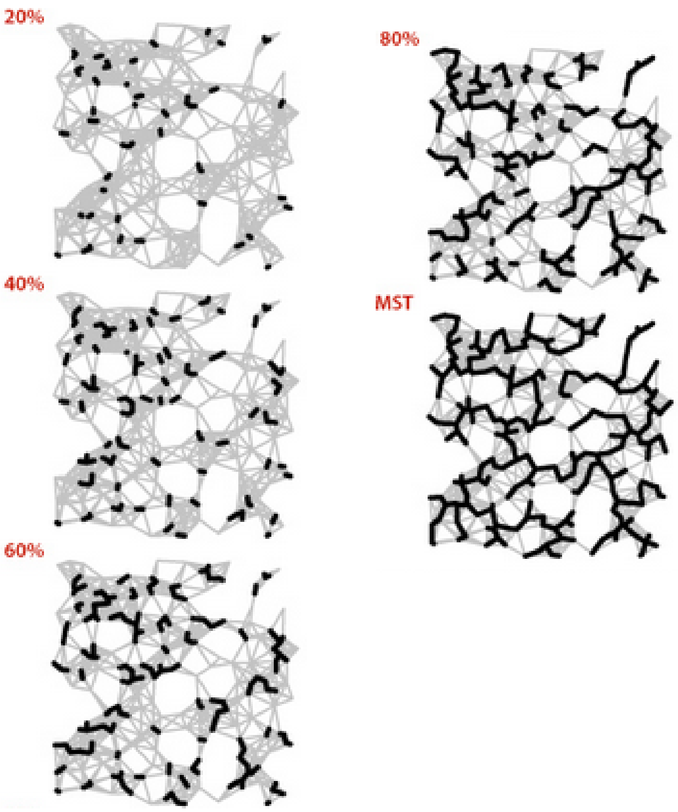
3. Enquanto ***|T| < |V| - 1*** faça

remova uma aresta ***e*** *de peso mínimo de* ***S*** *(Aqui está o método guloso)*

se ***e*** liga duas componentes distintas de ***T*** então adicione ***e*** à ***T***; Senão descarta ***e***

Algoritmo de Kruskal

- Resultado da Implementação



Algoritmo de Kruskal

- Análise do Algoritmo
  - O tempo de execução do algoritmo de Kruskal para um grafo G = (V, E) depende da implementação da estrutura de dados de conjunto disjuntos.
  - Vamos supor:
    - Tempo para ordenar as arestas O(ElogE)
    - União ou descarte das arestas O(E)
    - Cada vértice possui o numero de vertices – 1, com isso teremos que passar por O(lg V)
    - Conclui-se que o tempo de execução do algoritmo de kruskal é O(E log V).

Cormen, 2002

Algoritmo de Kruskal

- Análise do Algoritmo
  - O tempo de execução do algoritmo de Kruskal para um grafo G = (V, E) depende da implementação da estrutura de dados de conjunto disjuntos.
  - Vamos supor:
    - Tempo para ordenar as arestas O(ElogE)
    - União ou descarte das arestas O(E)
    - Cada vértice possui o numero de vertices – 1, com isso teremos que passar por O(lg V)
    - Conclui-se que o tempo de execução do algoritmo de kruskal é O(E log V).

Cormen, 2002

Agenda

- Aula anterior
- Introdução
- Método guloso
- Algoritmo de Kruskal
- Exercícios

Exercícios

- 1. Implemente os dois algoritmos e faça a análise de complexidade . Utilize o livro do Cormen.
  - Algoritmo de Kruskal:
    - http://stackoverflow.com/questions/8201354/c-implementation-of-kruskals-algorithm-for-mst
  - Algoritmo de Prim:
    - https://pt.wikipedia.org/wiki/Algoritmo_de_Prim#Implementa.C3.A7.C3.A3o_em_C

Exercícios

- L4

Agenda

- Aula anterior
- Introdução
- Método guloso
- Algoritmo de Kruskal
- Exercícios
- Próxima aula

Próxima aula

- Método probabilístico
  - Algoritmo de huffman

AULA 16

Prof. Mathias