Grafos e Algoritmos Computacionais

Algoritmo Guloso:
Determinação da Árvore Geradora Máxima

Prof. André Britto

Departamento de Computação/UFS

Algoritmo Guloso

- **Problema Geral**

  - Seja dado um conjunto S. Deseja-se determinar um subconjunto $S' \subseteq S$ tal que :

    (i) S' satisfaz uma dada propriedade P, e

    (ii) S' é o máximo (ou mínimo) em relação a algum critério α, que satisfaz P.

  - Ou seja, S' é o maior (ou menor) subconjunto de S, segundo α, que satisfaz P.

Algoritmo Guloso

- Algoritmo guloso resolve esse problema através de um processo iterativo para a construção de S' a partir de elementos de S.

- Seja S'_{i-1} o subconjunto assim construído após a iteração $i-1$. Na iteração i determina-se o elemento $s \in S - S'_{i-1}$, tal que:

  (i) $S'i\text{-}1 \cup \{s\}$ satisfaz P, e

  (*ii*) S é tal que $S'i\text{-}1 \cup \{s\}$ é maior (menor) ou igual, segundo α, do que $S'i\text{-}1 \cup \{r\}$ que satisfaz P, para todo $r \in S - S'i$.

Algoritmo Guloso

- Características do guloso:
  - a cada passo procure-se incorporar a S' a melhor porção de S;
  - pode não conduzir ao resultado exato do problema $\Rightarrow$ precisa provar que o resultado obtido é de fato máximo (mínimo);

Aplicação do Algoritmo Guloso

- Determinação da Árvore Geradora Máxima
  - Seja G um grafo conexo, em que cada aresta $a = (v,w)$ possui um peso $p(a)$. Denomina-se peso de uma árvore geradora T de G, a soma dos pesos das arestas de G que formam T, ou seja o peso de T é dado por:

$$p(T) = \sum_{a \in aT} p(a)$$

Árvore Geradora Máxima

- **Problema:**

  - Determinar um subconjunto $aT \subseteq EG$, tal que:
    (i) T seja uma árvore geradora de G, e
    (ii) $p(T)$ seja máximo.

  - Propriedade $P \Rightarrow T$ ser árvore geradora.
  - Critério $\alpha \Rightarrow$ peso de T seja máximo.

Árvore Geradora Máxima: Algoritmo de Kruskal

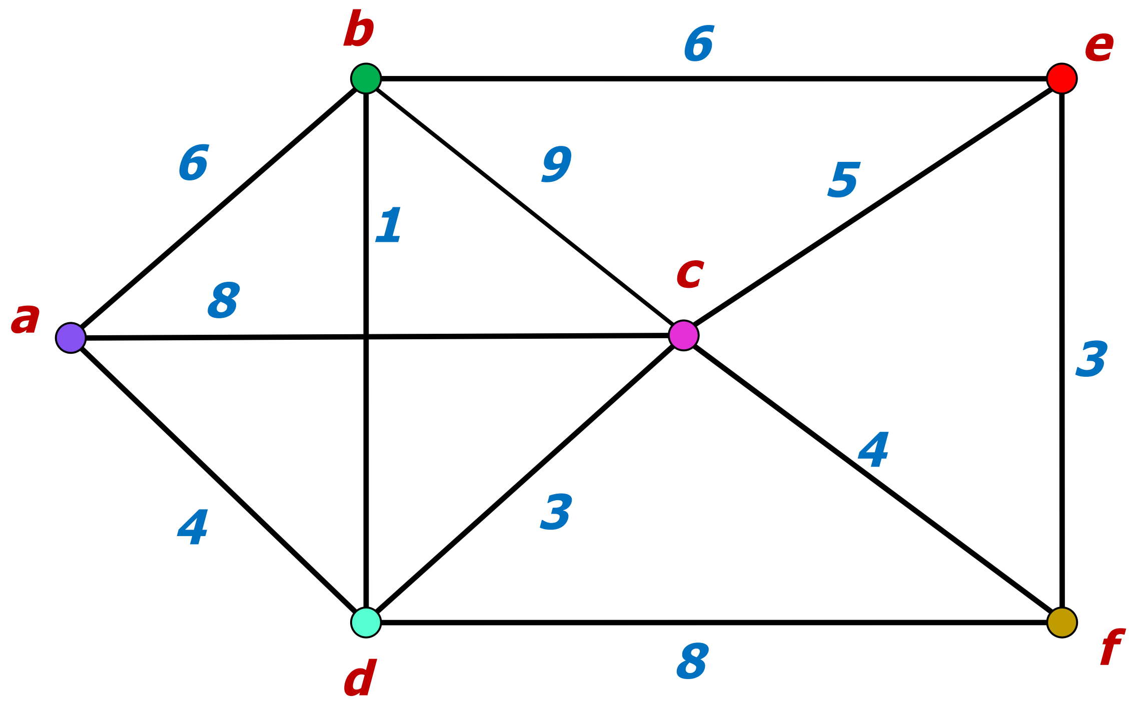
- **Guloso**
- Passos gerais:
  - definir inicialmente $aT = \emptyset$
  - a cada passo escolher uma aresta *(v,w)*, ainda não considerada tal que:

    (i) a incorporação de *(v,w)* ao conjunto *aT* até então obtido não gere ciclos.

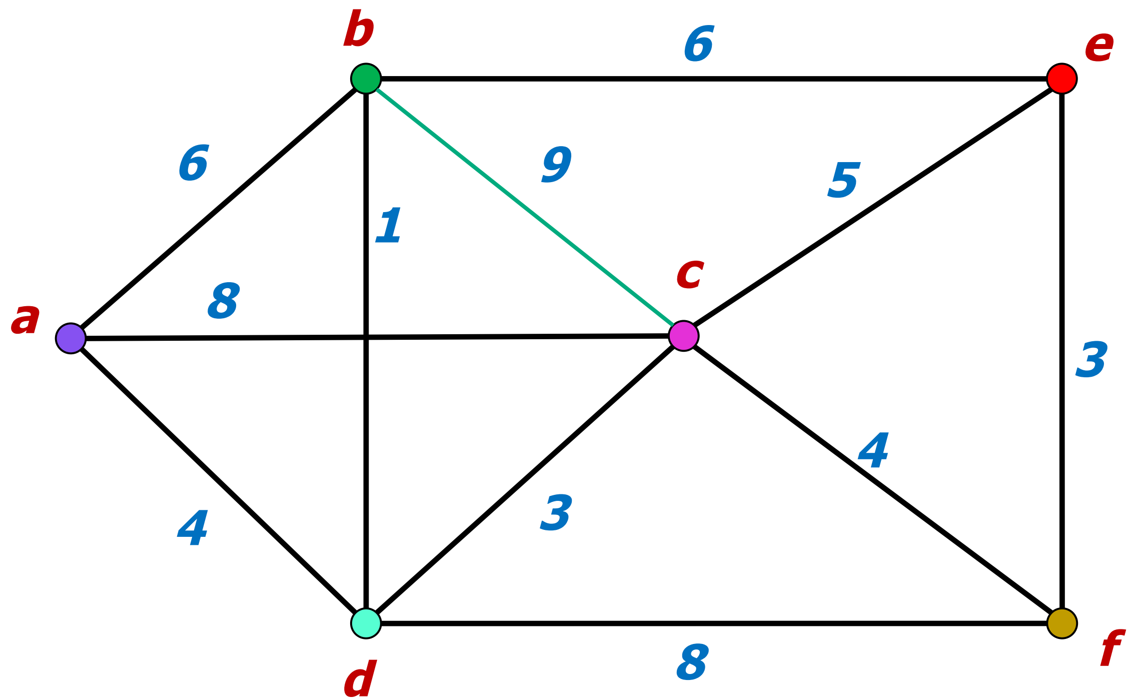
    (ii) o peso total de $aT \cup \{(v,w)\}$ seja máximo dentre todas as escolhas que satisfazem (i).

  - incorporar a aresta escolhida a *aT* e repetir o processo, até que todas as arestas tenham sido consideradas.

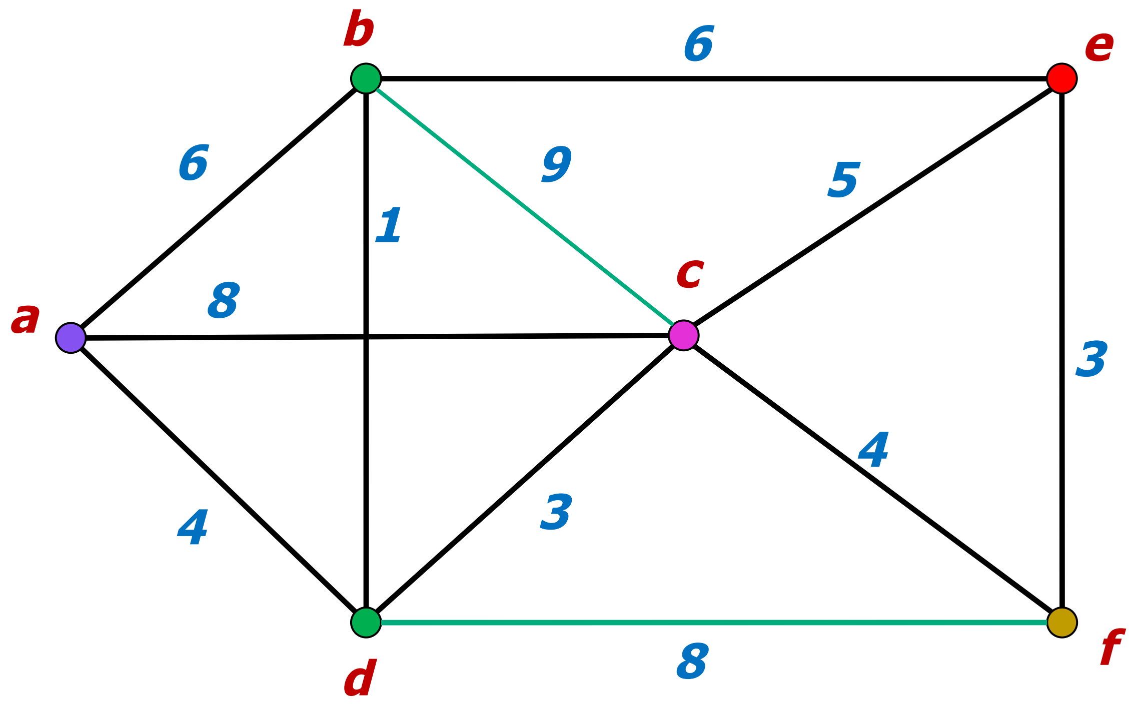
Algoritmo de Kruskal: Exemplo



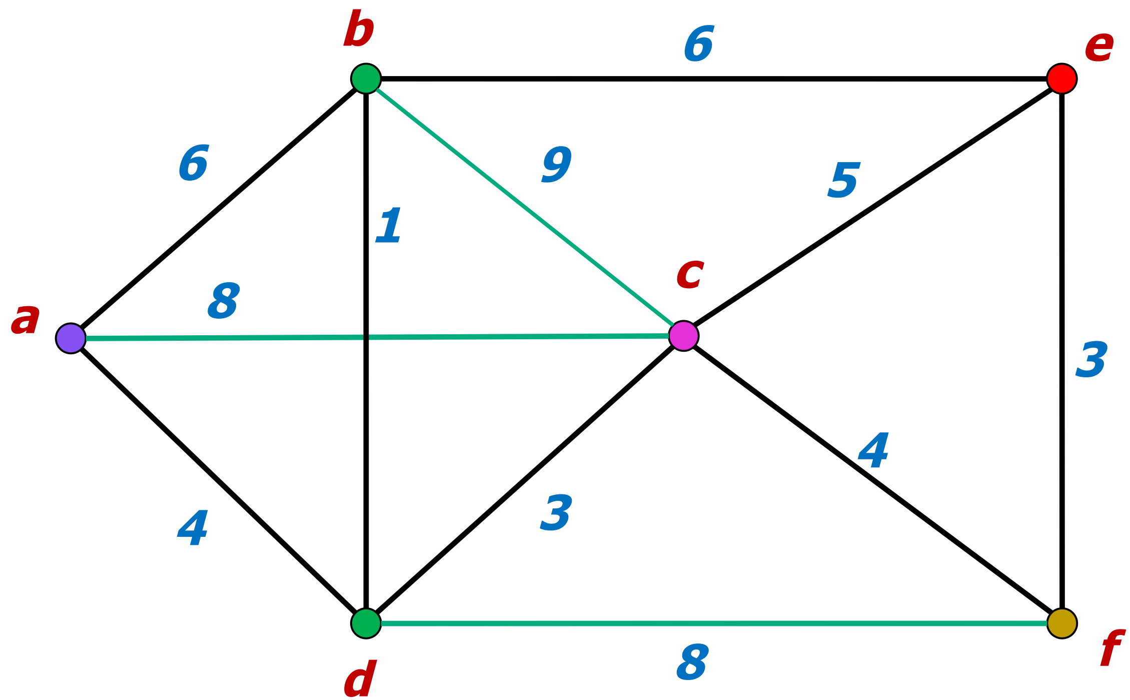
Algoritmo de Kruskal: Exemplo



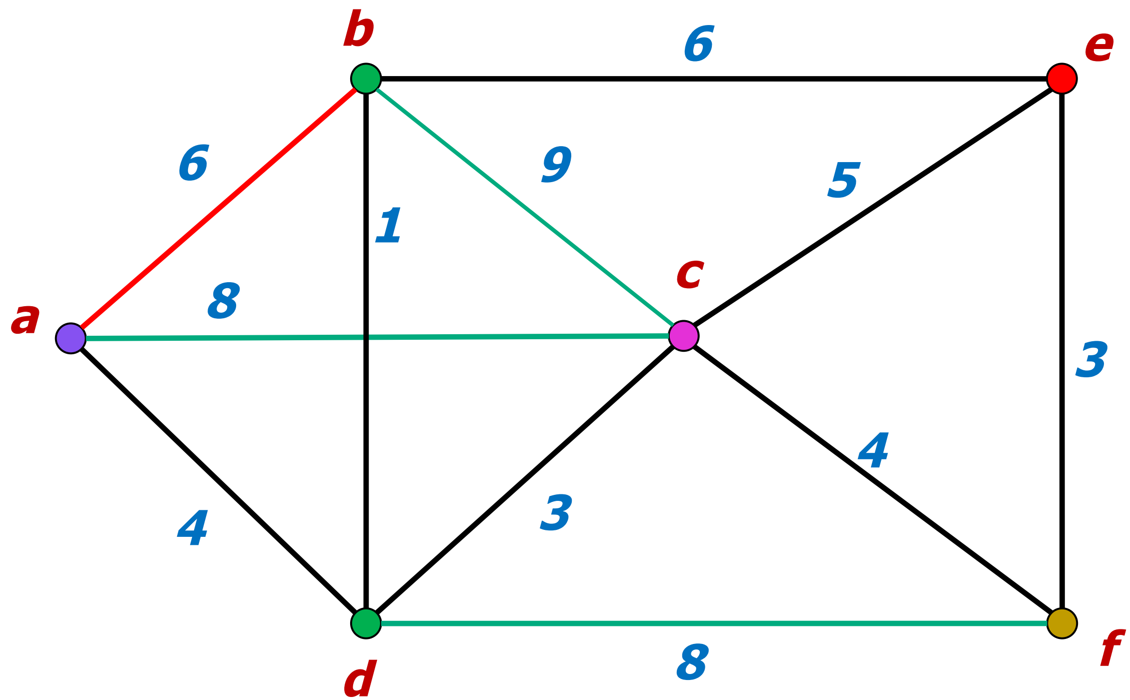
Algoritmo de Kruskal: Exemplo



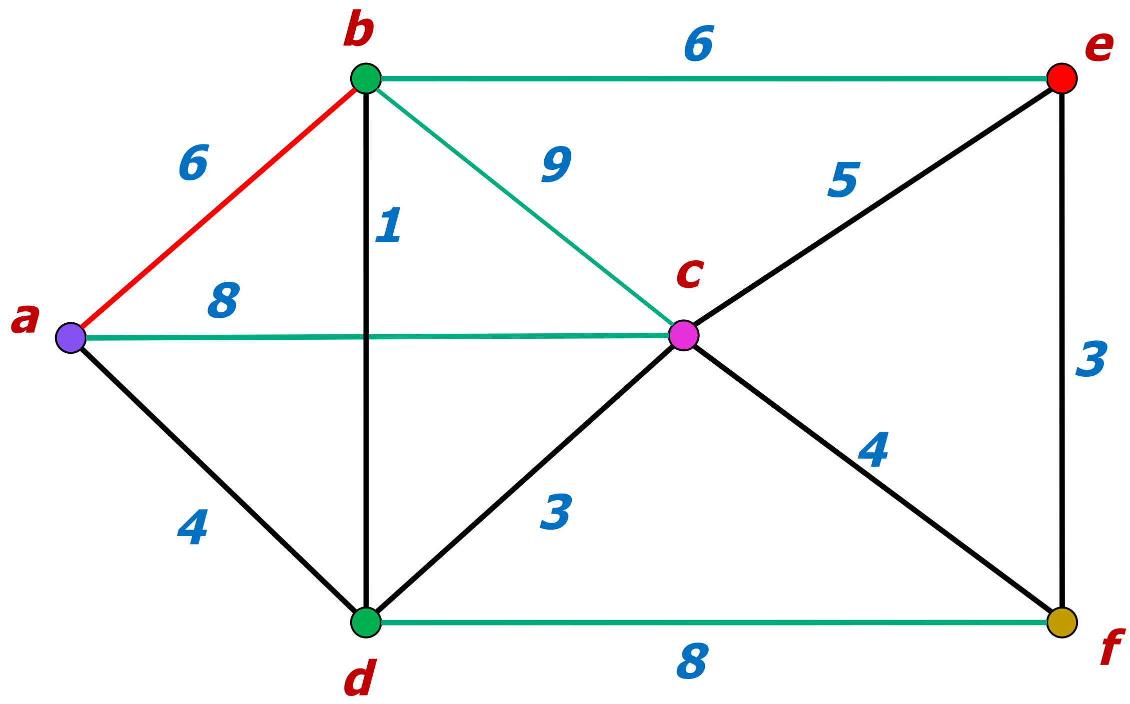
Algoritmo de Kruskal: Exemplo



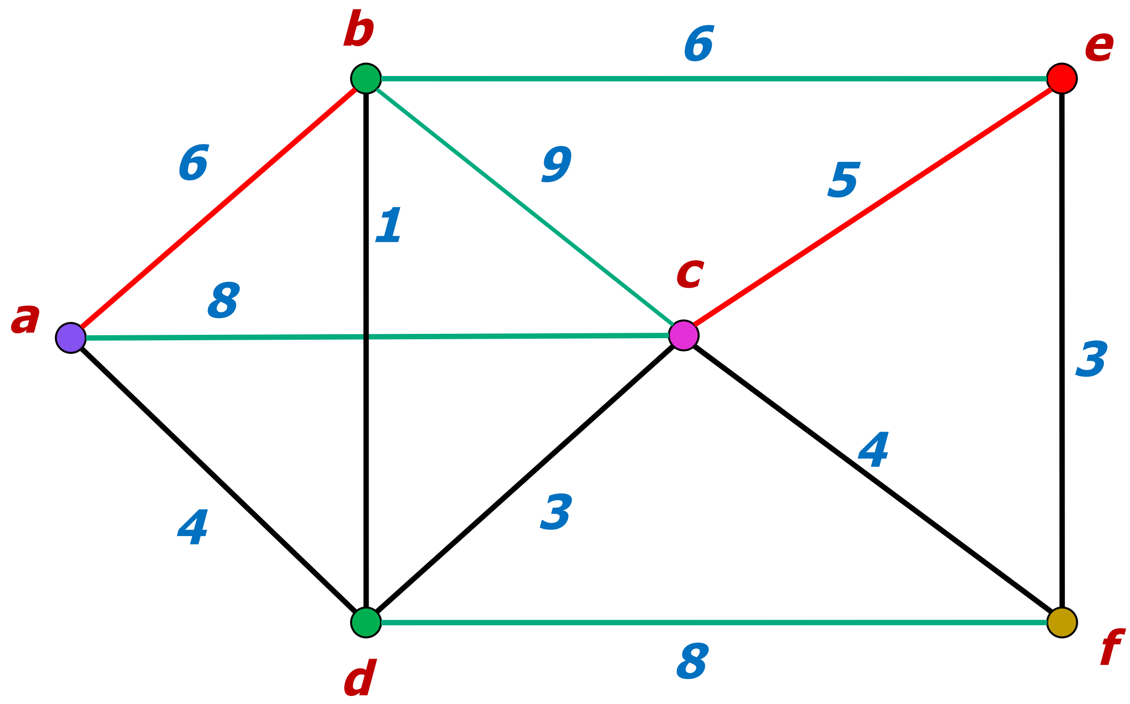
Algoritmo de Kruskal: Exemplo



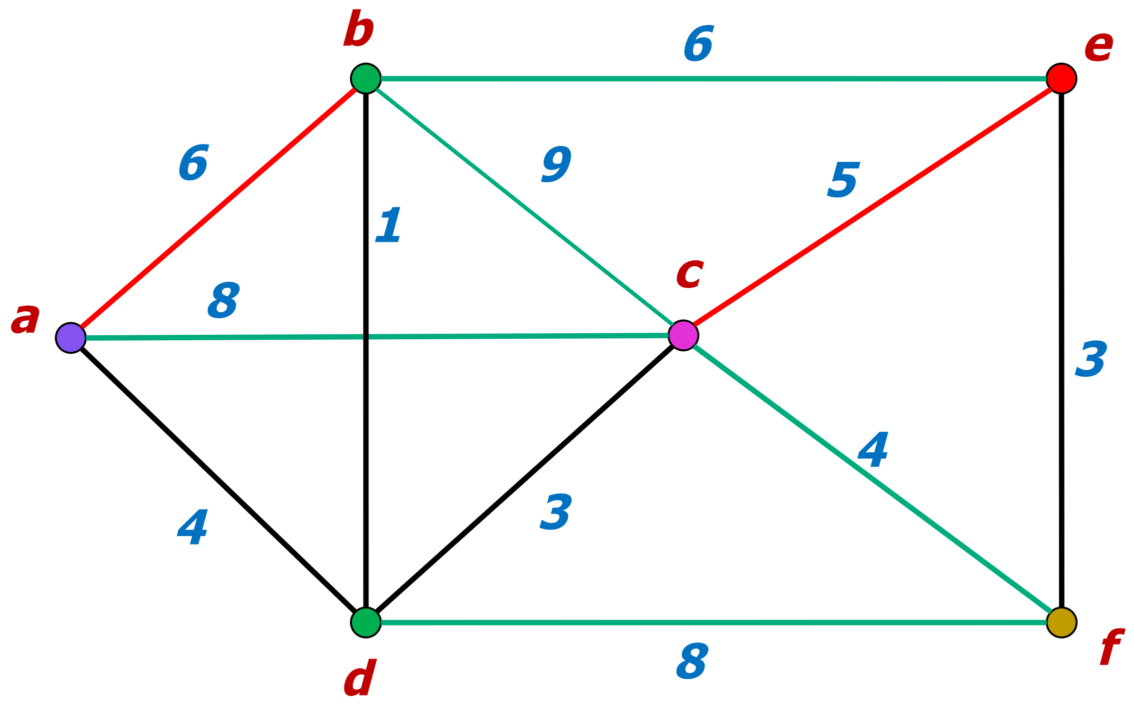
Algoritmo de Kruskal: Exemplo



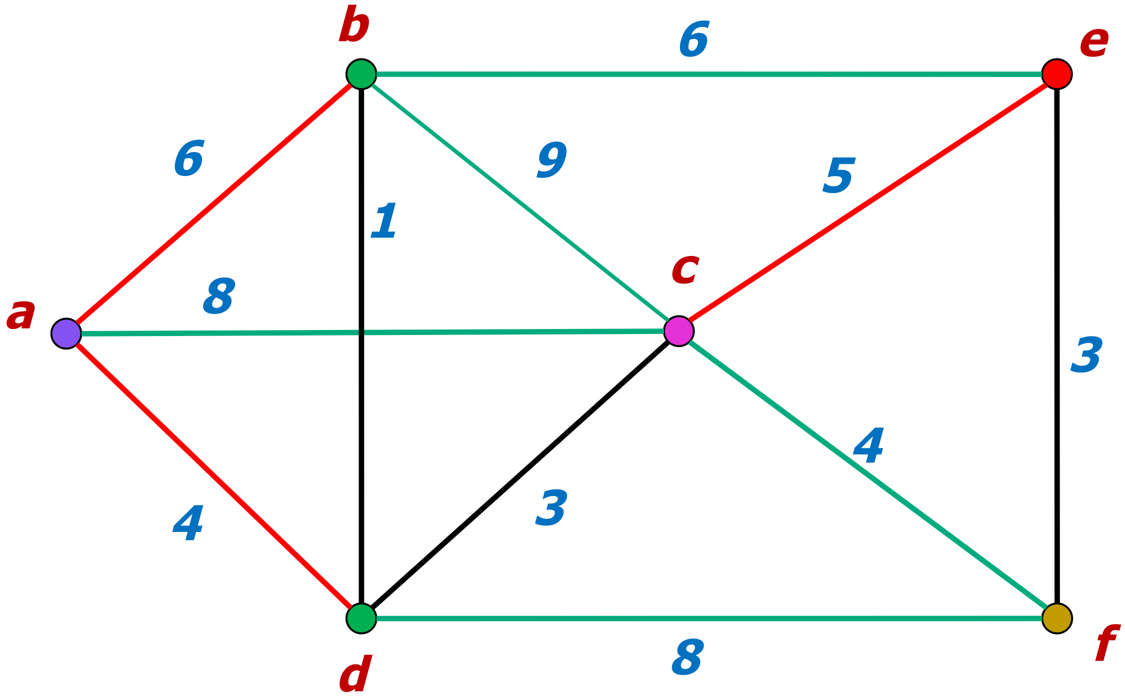
Algoritmo de Kruskal: Exemplo



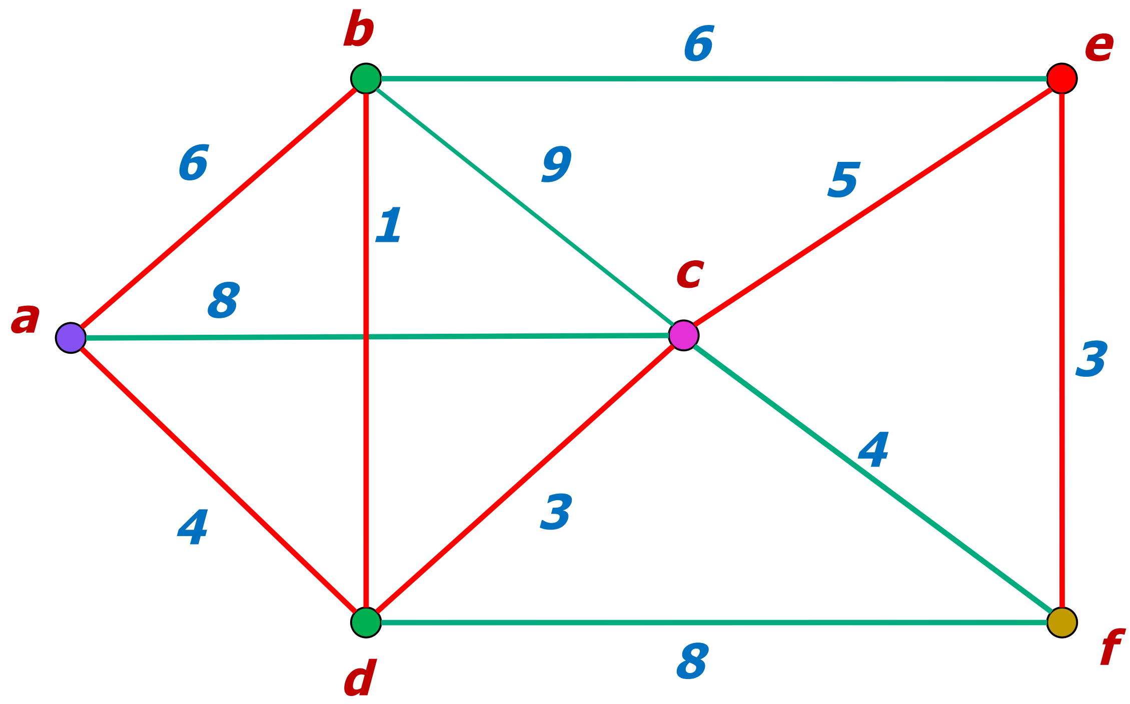
Algoritmo de Kruskal: Exemplo



Algoritmo de Kruskal: Exemplo



Algoritmo de Kruskal: Exemplo



Algoritmo de Kruskal

Utiliza uma estrutura de dados de conjuntos disjuntos para manter vários conjuntos disjuntos de elementos.

Ordena as arestas de acordo com o peso.

Combina as árvores encontradas por um procedimento de união

Union - Find

Algoritmo de Kruskal

```
algoritmo ArvoreGeradoraMaxima
{dados: grafo G conexo com n vértices e m arestas}
início
   para j = 1,2,...,n faça Sj := {vj};     Criação dos conjuntos
   aT := ∅;
   Sejam a1,a2,...,am as arestas de G ordenadas segundo pesos
  decrescente;          Ordenação das arestas
   para j = 1,2,...,m faça
     início
       Seja (v,w) o par de vértices extremos de aj;
       Se v e w pertencem respectivamente a conjuntos     FIND
       disjuntos Sp e Sq então
          início
             Sp := Sp ∪ {Sq};    UNION
             eliminar Sq;
             aT := aT ∪ {aj};
          fim
     fim
fim
```

Algoritmo de Kruskal

- Complexidade?

Algoritmo de Kruskal

```
algoritmo ArvoreGeradoraMaxima
{dados: grafo G conexo com n vértices}
início
   para j = 1,2,...,n faça Sj := {vj};
   aT := ∅;
   Sejam a1,a2,...,am as arestas de G ordenadas segundo pesos
  decrescente;
   para j = 1,2,...,m faça
     início
       Seja (v,w) o par de vértices extremos de aj;
       Se v e w pertencem respectivamente a conjuntos
       disjuntos Sp e Sq então
          início
            Sp := Sp ∪ {Sq};
            eliminar Sq;
            aT := aT ∪ {aj};
          fim
     fim
fim
```

O (m log m)

Executa m operações

UNION – log (n)

Algoritmo de Kruskal

- Complexidade?

$$O(m \log n)$$

Árvore Geradora Máxima

- O algoritmo pode ser considerado como a construção de uma árvore geradora a partir de uma floresta.

- **O conjunto *aT* construído pelo algoritmo forma de fato uma árvore geradora de *G*?**

Lema 1

Seja *G* um grafo conexo e *T* a árvore geradora obtida pela aplicação do algoritmo guloso. Então *T* é uma árvore geradora de *G*.

Árvore Geradora Máxima

- **Será que *p(T)* é de fato máximo ?**

Lema 2

Seja *G* um grafo conexo e *T* a árvore geradora obtida pela aplicação do algoritmo guloso. Então *T* possui peso máximo.

- Raciocínio análogo para árvore geradora de peso mínimo.

Referências

- Seções 5.2 e 5.3 do Szwarcfiter, J. L., *Grafos e Algoritmos Computacionais,* Ed. Campus, 1983.
- Seção 23 do Cormen, *Introduction to Algorithms,* MIT Press, 2001.

Adaptado do material da Profa. Leila Silva.

Seção 3.3 do Grafos: conceitos, algoritmos e aplicações. Goldbarg, E. e Goldbarg M. Elsevier, 2012