**Minimum Spanning Tree**

**Definição**

Dado um grafo não orientado, conexo e ponderado G = (V, A), a árvore geradora mínima é um subgrafo de G, denotado por G' = (V, A'), que conecta todos os vértices de G e com o menor peso total final.

Uma **árvore** é um grafo não orientado, conexo e acíclico.

**Algoritmos Usuais**

Dois algoritmos muito usuais para resolver o problema da árvore geradora mínima são os algoritmos de Kruskal e de Prim.

Ambos os algoritmos usam uma abordagem gulosa para resolver os problemas, em cada iteração os algoritmos “abocanham” a aresta que parece mais promissora naquele momento sem se preocupar com o efeito global dessa escolha. Apesar de utilizarem essa abordagem, é sempre garantido que se alcance a melhor solução possível. Além disso, as execuções dos algoritmos são muito rápidas.

**Método Genérico**

É feito um loop sobre as arestas de G e a cada passo é determinada uma aresta (u, v) que poderá ser inserida em G'. Essa aresta é chamada **aresta segura**, pois ela pode ser adicionada com segurança em G' sem que prejudique a formação de uma MST.

Cada um dos algoritmos possui uma definição própria do que considera uma **aresta segura.**

**Algoritmo de Kruskal**

O algoritmo de Kruskal considera uma **aresta segura**, uma aresta (u,v) que dentre todas as arestas de G tenha o menor peso (ou custo) e que já não tenha sido inserida.

Utiliza uma estrutura de dados auxiliar denominada **Disjoint-Set (Union Find)** para armazenar vários conjuntos disjuntos de vértices, isto é, inicialmente cada vértice é o líder do seu próprio conjunto (conjunto unitário).

**Pseudo-código:**

* G’ = vazio
* **Para** cada vértice v pertencente à G.V
  + Criar conjunto com v
* Ordenar as arestas de G.A em ordem crescente pelo peso p
* **Para** cada aresta (u, v) pertencente a G.A
  + **Se** o conjunto de u for diferente do conjunto de v
    - Adicionar a aresta em G’
    - Unir os conjuntos (u, v)

**Algoritmo de Prim**

O algoritmo de Prim considera uma **aresta segura**, uma aresta (u, v) ∈ A que possua o menor peso (custo) e que seja incidente a um vértice que já pertença a árvore, que ainda não tenha sido inserida e não forme ciclo.

**Pseudo-código:**

* **Para** cada vértice u pertencente a G.V
  + u.chave = infinito //menor custo entre as arestas incidentes no vértice u
  + u.pai = nulo //vértice anterior
* Fila de prioridade mínima Q = G.V
* **Enquanto** Q != nulo
  + u = retirarMenor(Q)
  + **Para** cada vértice v pertencente a G.Adj[u]
    - **Se** o vértice v pertence a Q **e** peso(u, v) < v.chave
      * v.pai = u
      * v.chave = peso(u, v)