

Marlon Fernandes de Alcantara
Ata referente ao algoritmo de Kruskal
29 de março de 2012

Em um grafo conexo a árvore geradora mínima conterá exatamente $n-1$ arestas, onde n é o número de vértices. O algoritmo de **Kruskal** parte apenas dos vértices, considerando como se fosse uma floresta onde cada vértice pertence a uma árvore isolada; então são inseridas as menores arestas até que se consiga conectar todos os vértices; para isso, incluindo apenas arestas que reduzem o número de componentes do grafo.

O algoritmo de **Kruskal** pode ser descrito em pseudocódigo como segue:

```
// O algoritmo recebe uma lista de arestas da forma (u, v, w) mostrando que existe uma aresta que liga u,v  
(logo v,u também) com peso w, e retorna uma lista de arestas que pertencem a árvore geradora mínima.
```

```
arestas Kruskal(vértices V, arestas E){  
    ordenarCrescente(E) // ordenar pelo campo w  
    para cada vértice v em V: v pertence a uma sub-árvore distinta  
    arestas K // as arestas finais pertencentes a árvore mínima serão inseridas aqui  
    para cada aresta e em E:  
        se sub-árvore(e.u) != sub-árvore(e.v):  
            K inclui e  
            sub-árvore de e.u e de e.v à partir de agora são a mesma  
    retorne K  
}
```

O algoritmo de **Kruskal** é um dos algoritmos gulosos ótimos. Como não existe um critério de desempate para a escolha das arestas, caso haja mais de uma árvore geradora mínima, qualquer uma delas pode ser dada como resposta.

Segue em anexo uma implementação do **Kruskal** em C++ utilizando o **UnionFind** para distinguir as diferentes sub-árvores.