

Atividade por: Kaiky França da Silva Matrícula: 802169

Algoritmo de Kruskal

O algoritmo de Kruskal pode ser utilizado para encontrar a MST (Maximal Spanning Tree), para isso as arestas do grafo ponderado serão ordenadas por ordem de peso em ordem crescente e será inicializado um conjunto de componentes conexos para os vértices e vai ser percorrida toda a lista de arestas, e serão adicionadas à árvore se não forem ciclos, até que todos os vértices estejam conectados.

Para encontrar a árvore geradora máxima, basta apenas modificar um simples passo que é a ordenação das arestas, ao seja, ao invés de ordenar as arestas por seus pesos em ordem crescente, será ordenado pela ordem decrescente, e os demais passos do algoritmo será permanecido o mesmo, e assim será encontrada a Árvore Geradora Máxima.

A complexidade do algoritmo de Kruskal é $O(E \log E)$, onde E é o número de arestas. Como a única modificação é a ordenação em ordem inversa, a complexidade não se altera.

Algoritmo de Prim

O algoritmo de Prim constrói a MST a partir de um único vértice e expande a árvore escolhendo a aresta de menor peso disponível, logo os passos do algoritmo seriam a inicialização de um conjunto de vértices já incluídos na árvore, e será utilizado uma fila de prioridade (normalmente um heap mínimo) para escolher a aresta de menor peso conectando um vértice já na árvore a um vértice fora dela.

As modificações necessárias para encontrar a árvore geradora máxima, basta modificar a maneira como é escolhida as arestas, ao invés de utilizar um heap mínimo (que irá priorizar as arestas pelo menor peso) será utilizado um heap máximo (onde a prioridade vai ser as arestas de maior peso)

A complexidade do algoritmo de Prim, quando implementado com um heap binário, é $O(E \log V)$, onde V é o número de vértices. Como a única alteração é o uso de um heap máximo em vez de um heap mínimo, a complexidade não se altera.