Yusuf Fawzy Elnady

Kruskal's Minimum Spanning Tree Algorithm

(Greedy - Disjoint Sets)

Minimum Spanning Tree:

Given a **connected undirected graph**, a **spanning tree** of that graph is a **subgraph** that is a **tree** and **connects all the vertices together**. A **single graph** can have **many different spanning trees**. A **minimum spanning tree** (MST) or minimum weight spanning tree for a weighted connected undirected graph is a spanning tree with **the minimum possible weight**. The **weight** of a spanning tree is the **sum of weights** given to each edge included in the spanning tree.

Kruskal's Algorithm:

The algorithm initially considers each node as **isolated tree** using the **disjoint sets data structure**, it then proceeds to process the edges in the ascending order of their costs/weights. For each edge, it is **ignored**, if it **connects two nodes in the same tree**, or it is **included**, if it connects **two nodes in different trees**, and then it **joins** these two trees using the **union operation** of the disjoint sets data structure.
If we included an edge that connects two nodes in the same tree, it will form a **cycle**, and the resulting subgraph will not be a tree. So basically, the algorithm starts with **multiple MSTs**, and proceed to include edges until **all nodes** are in the **same tree,** and we have a **single MST**.
Kruskal's algorithm is a **greedy** algorithm. The greedy choice is to pick the smallest weight edge that does not cause a cycle in the MST constructed so far.
Following is a pseudocode:

```
sort all edges in ascending order according to their weights
for each edge:
  if the edge connects two nodes in different trees
    count the edge in the total MST weight
    join the two trees of the two nodes on the edge under consideration
```

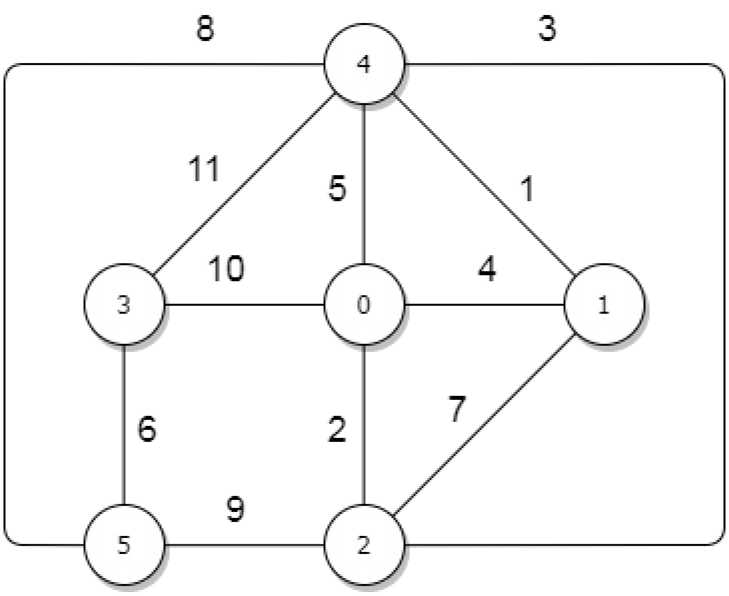
The project is built using **C++, Visual Studio 2019.**

Yusuf Fawzy Elnady

Example:

Let graph G contain 6 nodes and 11 undirected edges. An edge (a,b,w) connects nodes a and b with weight w. Graph G contains the following edges: {(0,1,4), (0,2,2), (0,3,10), (0,4,5), (1,2,7), (1,4,1), (2,4,3), (2,5,9), (3,4,11),(3,5,6), (4,5,8)}. Write the edges of the MST of G in the order obtained by Kruskal algorithm.

Sorting the edges in ascending order: (1,4,1), (0,2,2), (2,4,3), (0,1,4), (0,4,5), (3,5,6), (1,2,7), (4,5,8), (2,5,9), (0,3,10), (3,4,11). The algorithm will initialize the disjoint sets data structure with 6 nodes, the following table shows which edges will be included, and the disjoint sets data structure state after processing each edge:



Edge	State after	Included ?
(1,4,1)	{0}, {1, 4}, {2}, {3}, {5}	YES
(0,2,2)	{0, 2}, {1, 4}, {3}, {5}	YES
(2,4,3)	{0, 1, 2, 4}, {3}, {5}	YES
(0,1,4)	{0, 1, 2, 4}, {3}, {5}	NO
(0,4,5)	{0, 1, 2, 4}, {3}, {5}	NO
(3,5,6)	{0, 1, 2, 4}, {3, 5}	YES
(1,2,7)	{0, 1, 2, 4}, {3, 5}	NO
(4,5,8)	{0, 1, 2, 3, 4, 5}	YES
(2,5,9)	{0, 1, 2, 3, 4, 5}	NO
(0,3,10)	{0, 1, 2, 3, 4, 5}	NO
(3,4,11)	{0, 1, 2, 3, 4, 5}	NO

The edges: (1,4), (0,2), (2,4), (3,5), (4,5)

Total MST cost: 1 + 2 + 3 + 6 + 8 = 20