Program Structures and Algorithms

Final Project Piece

Topic – Kruskal's Algorithm

Authored by –

Name - Abhishek Chaudhari

NUID - 002928364

Guided by –

Prof. Nik Brown

Sai Vennela Nikkanti(TA)

Introduction –

The discrete mathematics field of graph theory introduces the idea of Kruskal's algorithm. A connected weighted graph's shortest path between any two locations is found using this method. For a given graph, a minimal spanning tree is created using the Kruskal algorithm. But what exactly is a Minimum Spanning Tree? A subset of a graph called a minimum spanning tree has edges equal to the number of vertices -1 and the same number of vertices as the original graph. For the total of all edge weights in a spanning tree, it likewise has a low cost.

Only if the selected edge does not form a cycle does Kruskal's method keep adding nodes to the tree. It sorts all edges in ascending order of their edge weights. Additionally, it chooses the edge with the lowest cost first and the highest cost last. In order to discover the global optimal solution, the Kruskal algorithm thus makes a locally optimal decision. Because of this, it is known as a greedy algorithm.

Let us have a look at the definitions.

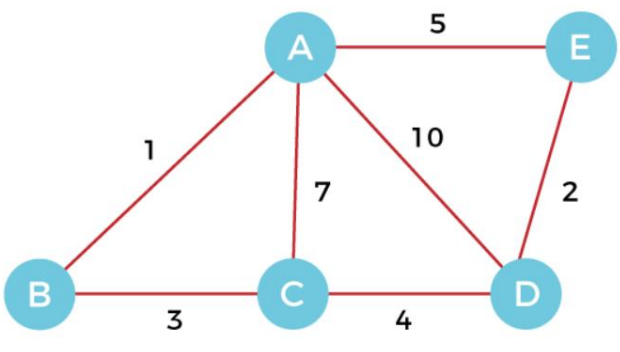
A. Minimum Spanning Tree - A minimum spanning tree or minimum weight spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

B. Greedy Algorithm - A greedy algorithm is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage. In many problems, a greedy strategy does not produce an optimal solution, but a greedy heuristic can yield locally optimal solutions that approximate a globally optimal solution in a reasonable amount of time.

C. Minimum Cost Edge: If the minimum cost edge of a graph is unique, then this edge is included in any MST. If a new edge is added to the spanning tree then it will become cyclic because every spanning tree is minimally acyclic.

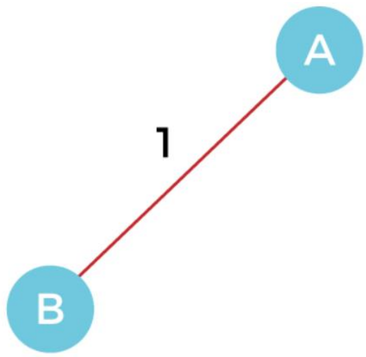
Steps for Kruskal's Algorithm –

Now let us understand the Algorithm with an example.

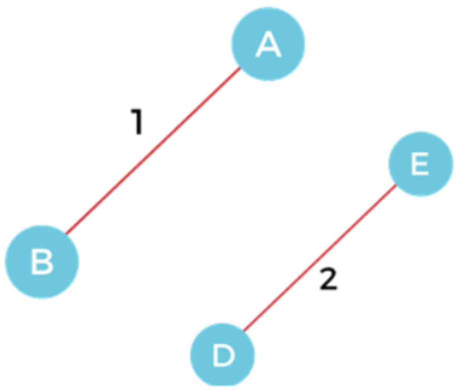


We need to find Minimum Spanning tree for the above given graph.

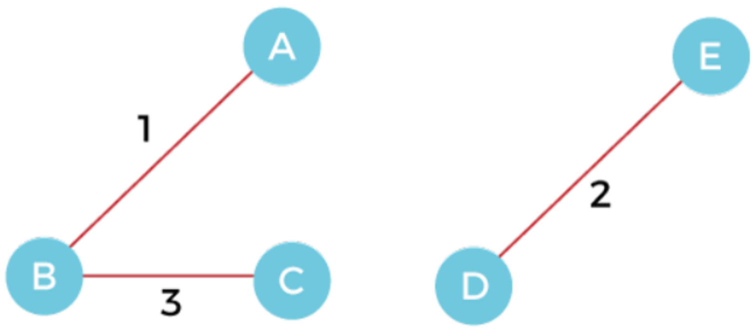
Step 1 - First, add the edge **AB** with weight **1** to the MST.



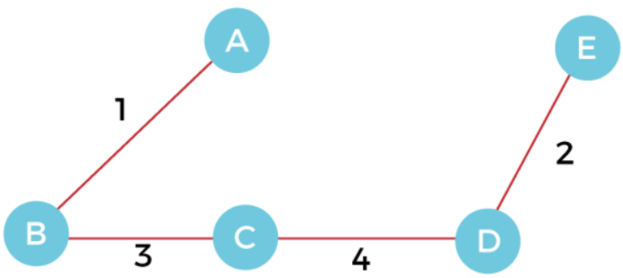
Step 2 - Add the edge **DE** with weight **2** to the MST as it is not creating the cycle.



Step 3 - Add the edge **BC** with weight **3** to the MST, as it is not creating any cycle or loop.



Step 4 - Now, pick the edge **CD** with weight 4 to the MST, as it is not forming the cycle.

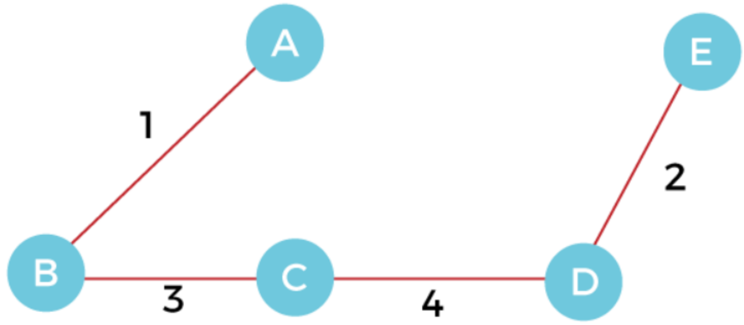


Step 5 - After that, pick the edge **AE** with weight **5.** Including this edge will create the cycle, so discard it.

Step 6 - Pick the edge **AC** with weight **7.** Including this edge will create the cycle, so discard it.

Step 7 - Pick the edge **AD** with weight **10.** Including this edge will also create the cycle, so discard it.

So, the final minimum spanning tree obtained from the given weighted graph by using Kruskal's algorithm is –



The cost of the MST is = AB + DE + BC + CD = 1 + 2 + 3 + 4 = 10. Now, the number of edges in the above tree equals the number of vertices minus 1. So, the algorithm stops here.

Pseudo Code of Kruskal's Algorithm –

KRUSKAL(G,w)

1.A = NULL.
2.for each vertex v in G.V
MAKE-SET(v)
1.Sort the edges G.E into non-decreasing order by weight w
2.For each edge (u,v) in G.E, taken in non decreasing order
a.If FIND-SET(u) != FIND-SET(v)
A= AU{(u,v)}
UNION(u,v)
3.Return A

Complexity of Kruskal's algorithm –

The time complexity of Kruskal's algorithm is O(E logE) or O(V logV), where E is the no. of edges, and V is the no. of vertices.

Applications of Kruskal's Algorithm -

The best example of Kruskal's Algorithm is Google Maps. It uses the Algorithm to find the cheapest path of travelling from source to destination.

Other Applications of the Algorithm are –

A. Mobile Network
B. Salesman
C. GPS
D. Landline cable
E. TV cable
F. DishTV Netwok

GitHub Link - https://github.com/chaudhari-ab/PSA_Final_Portfolio_Piece.git