TITLE: Implement Greedy search algorithm for Kruskal's Minimal Spanning Tree

Algorithm **AIM:** To apply Greedy search algorithm mechanism in Kruskal's MST.

OBJECTIVES: Based on above main aim following are the objectives

1. To study Greedy Search algorithm
2. To study Kruskal's minimum spanning tree
3. To determine complexity of algorithms

INTRODUCTION

Greedy Algorithms:

Greedy is an algorithmic paradigm that builds up a solution piece by piece, always choosing the next piece that offers the most obvious and immediate benefit. So the problems where choosing locally optimal also leads to global solution are best fit for Greedy.

Structure of a Greedy Algorithm:

Greedy algorithms take all of the data in a particular problem, and then set a rule for which elements to add to the solution at each step of the algorithm. In the animation above, the set of data is all of the numbers in the graph, and the rule was to select the largest number available at each level of the graph. The solution that the algorithm builds is the sum of all of those choices.

Kruskal's Minimum Spanning Tree Algorithm:

What is Minimum Spanning Tree?

Given a connected and 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 a weight less than or equal to the weight of every other spanning tree. The weight of a spanning tree is the sum of weights given to each edge of the spanning tree.

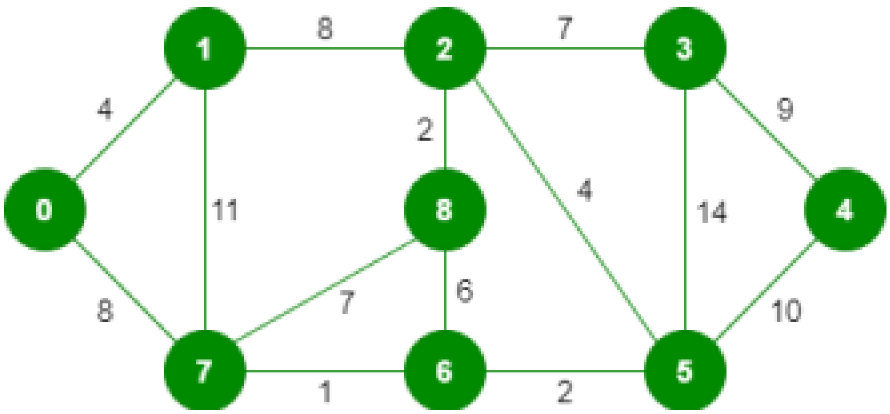
How many edges does a minimum spanning tree has? -

A minimum spanning tree has (V – 1) edges where V is the number of vertices in the given graph.

Below are the steps for finding MST using Kruskal's algorithm

1. Sort all the edges in non-decreasing order of their weight.
2. Pick the smallest edge. Check if it forms a cycle with the spanning tree formed so far. If cycle is not formed, include this edge. Else, discard it.
3. Repeat step 2 until there are (V-1) edges in the spanning tree.

The 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. Let us understand it with an example: Consider the below input graph.



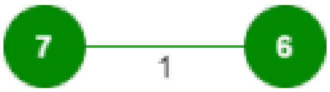
The graph contains 9 vertices and 14 edges. So, the minimum spanning tree formed will be having (9 – 1) = 8 edges.

```
After sorting:
```

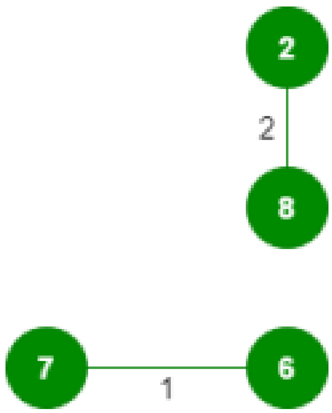
```
Weight Src Dest
1 7 6
2 8 2
2 6 5
4 0 1
4 2 5
6 8 6
7 2 3
7 7 8
8 0 7
8 1 2
9 3 4
10 5 4
11 1 7
14 3 5
```

Now pick all edges one by one from the sorted list of edges

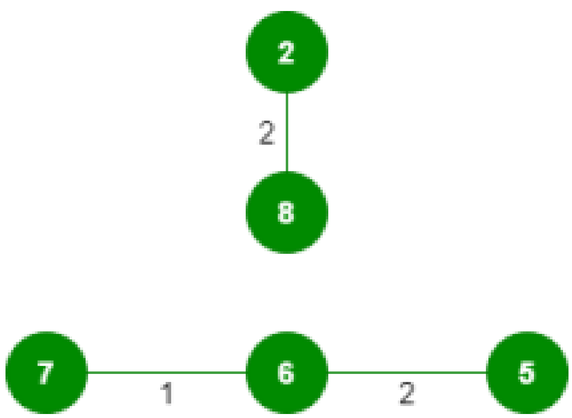
1. Pick edge 7-6: No cycle is formed, include it.



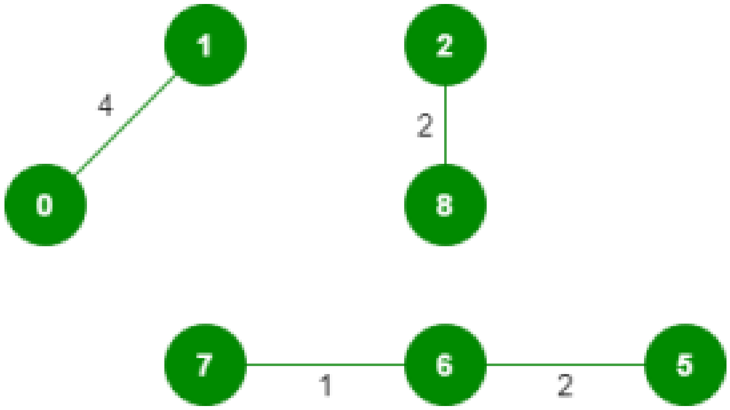
2. Pick edge 8-2: No cycle is formed, include it.



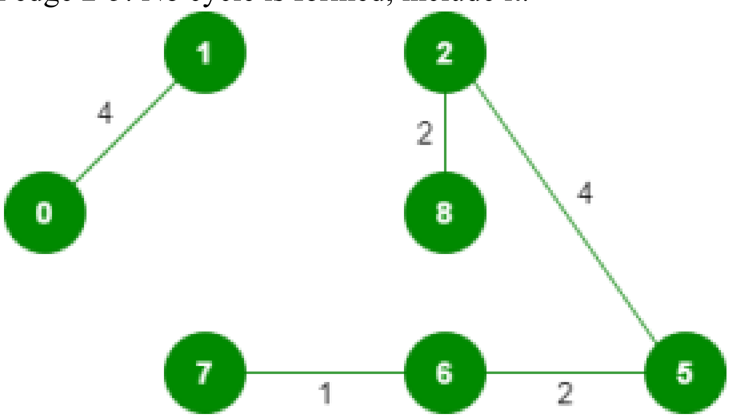
3. Pick edge 6-5: No cycle is formed, include it.



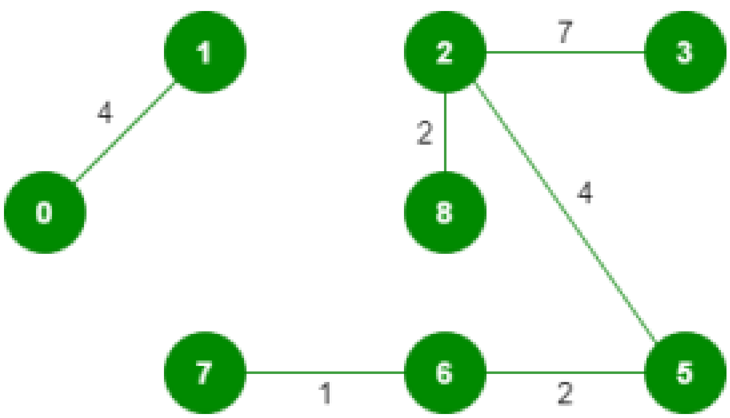
4. Pick edge 0-1: No cycle is formed, include it.



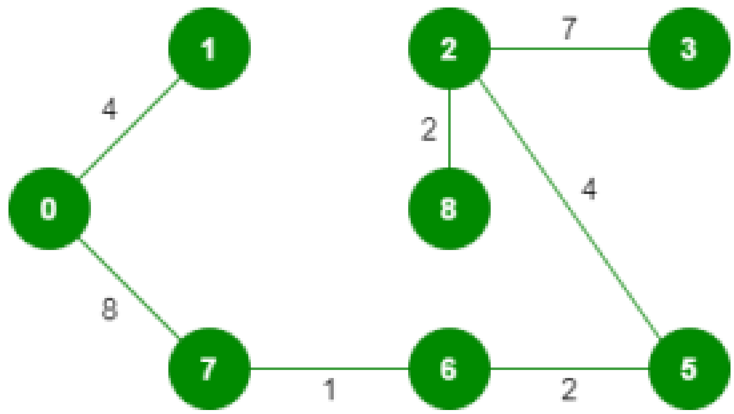
5. Pick edge 2-5: No cycle is formed, include it.



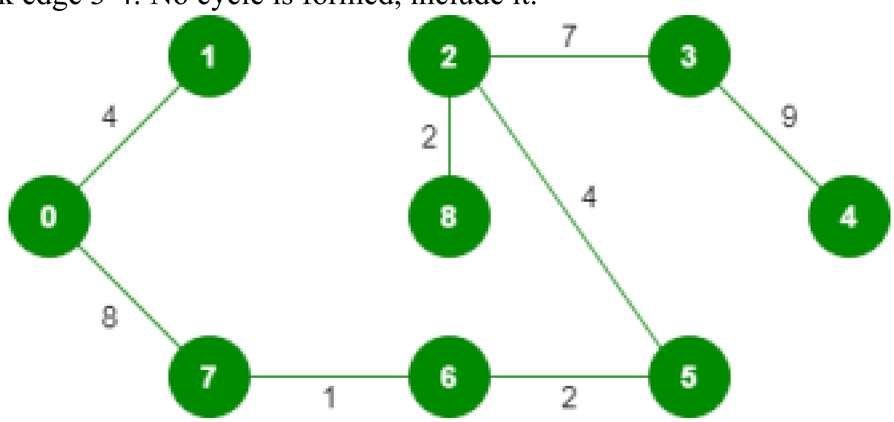
6. Pick edge 8-6: Since including this edge results in the cycle, discard it. 7. Pick edge 2-3: No cycle is formed, include it.



8. Pick edge 7-8: Since including this edge results in the cycle, discard it. 9. Pick edge 0-7: No cycle is formed, include it.



10. Pick edge 1-2: Since including this edge results in the cycle, discard it. 11. Pick edge 3-4: No cycle is formed, include it.



Since the number of edges included equals (V – 1), the algorithm stops here.

Time Complexity:

O(ElogE) or O(ElogV). Sorting of edges takes O(ELogE) time. After sorting, we iterate through all edges and apply the find-union algorithm. The find and union operations can take at most O(LogV) time. So overall complexity is O(ELogE + ELogV) time. The value of E can be at most O(V2), so O(LogV) is O(LogE) the same. Therefore, the overall time complexity is O(ElogE) or O(ElogV)

Conclusion: Thus we have studied how to implement Greedy Search Algorithm in Kruskal's Minimum Spanning Tree.