Minimum Spanning Tree

Minimum Spanning Tree is basically the tree which have minimum cost to cover all the vertices of weighted connected graph. There are several algorithms to find Minimum Spanning Tree for particular given graph. Most of used Algorithms are following

- Kruskal's Algorithm
- Prim's Algorithm

This Document contains links of tutorials for above algorithms and some problems related to this. Try to solve this problems after understanding the tutorials very well. And feel free to ask any doubts.

Link To The Tutorial :

https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/tutorial/

Implementation :

https://www.geeksforgeeks.org/greedy-algorithms-set-2-kruskals-minimum-spanning-tree-mst/

https://www.geeksforgeeks.org/greedy-algorithms-set-5-prims-minimum-spanning-tree-mst-2/

Link To The Problems :

http://codeforces.com/contest/160/problem/D (Must do question)

https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/practice-problems/algorithm/mr-president/

https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/practice-problems/algorithm/efficient-network/

https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/practice-problems/algorithm/3-types/

https://www.hackerearth.com/practice/algorithms/graphs/minimum-spanning-tree/practice-problems/algorithm/ioi-2050-2/

https://www.hackerearth.com/problem/algorithm/5-7/

Strongly Connected Components

Connected Graph:

In undirected graph, each node connected to every node by some path then this graph is called connected graph. We can find given graph is connected or not using simple DFS or BFS starting from any point.
If DFS or BFS include all nodes then given graph is connected otherwise not connected.

Strongly connected Graph:

Strongly connected Graph is similar as connected graph but this term use for directed graph. If we can reach every node from each node in directed graph then this graph is called strongly connected graph.

If we try to apply BFS/DFS from every vertex then it will take very much time and give TLE. So we must apply some different algorithm for that.

Links To The Tutorials:

https://www.hackerearth.com/practice/algorithms/graphs/strongly-connected-components/tutorial/

https://www.geeksforgeeks.org/strongly-connected-components/

Problems:

https://www.hackerrank.com/contests/world-codesprint-11/challenges/hackerland

https://www.hackerearth.com/practice/algorithms/graphs/strongly-connected-components/practice-problems/algorithm/a-walk-to-remember-qualifier2/

TOPOLOGICAL SORT

Tutorial:

https://www.hackerearth.com/practice/algorithms/graphs/topological-sort/tutorial/

Implementation

https://www.geeksforgeeks.org/topological-sorting/

Problems:

https://www.geeksforgeeks.org/maximum-edges-can-added-dag-remains-dag/