➔A Short Description of Prim's Algorithm

Prim's algorithm is a shortest path/pathfinding algorithm that developed by Robert C. Prim.

Algorithm can described as;

- ➔ Initialize a empty tree.
- ➔ Select a random vertex and add to tree.
- ➔ Go shortest edge destination vertex and add to tree if it is not in the tree.
- ➔ Repeat step 3 until all vertices in tree.

➔Big Oh analysis of Prim's algorithm and Kruskal's algorithm

In Prim's algorithm we constantly searching the shortest edge which is not in the tree until all vertices in the tree.

If use matrix then we going to find shortest edges in T(n) and we need to repeat it until get all vertices which is also T(n), n is number of vertices. We know they are nested loop so we can easily say it worst case complexity is O(n^2).

In Kruskal's algorithm first we sort after that removing duplicates and then iterate through all edges and apply the union-find. Union find take O(logV) in worst case. In many application (not mine) sorting also takes O(ElogE) or linear time O(N) in final

O(ElogE + logV) = O(ELogE) or O(sorting time + edge taversel * (Merging Operation + find operation))

Best Complexity is: O(ElogE + logV) = O(ELogE)

➔Performance comparison results between Kruskal's algorithm and Prim's algorithm

In my pc I could get precise results, but I sharing with you.

Prim's: 0.031 sec

Kruskal's; 0.032 sec