Div A Batch - C

David Daniella 20233000388

This implementation of kruskalls and prim's algorithm has been done in C++. Both of them are greedy algorithms. In kruskals, we put all edges in a priority queue and then pop and add to tree if it does not violate loop. In prims we go in order starting from any node (can be the edge with least weight also) and use priority queue to take smallest weight.

To check if there is a loop in tree or not. DSA F structure is used. In this, parent rank method has been used. Rank array has been used to reduce the number of recursive calls in the find function. Instead of having worst possible O(n) time, it is O(1) time.

The time complexity of prim's algorithm is ~~O(V²)~~ O(E log V) where v is number of vertices and ~~sp~~ space complexity is O(V+E) for the priority queue. ~~Best case~~

The time complexity of kruskal's algo is O(E log E) and space complexity is O(V+E) only because of priority queue

From the graph plotted, we can clearly see that running time of kruskal's algo is less than prims. This is because in kruskalls, we simply directly look for optimal solution in priority queue. But in prims we must add edges then it's neighbours, then check, etc.