ASSIGNMENT - 02

COMP 544 - ALGORITHM

ALGORITHM ASSIGNMENT GROUP - 03

1. **Compare what happens in Kruskal's algorithm versus Prim's algorithm if we run them on a graph that is not connected.**

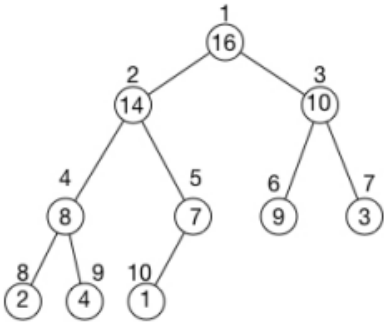
Ans. A spanning forest, or collection of spanning trees, each of which is an MCST for each linked component in the original, disconnected graph, is what is produced when Kruskal's algorithm is applied to a disconnected graph. Due to the fact that Kruskal's takes into account edges and only adds an edge if it does not result in a cycle, this is the case. Since there is no scenario in which a cycle can be produced between connected components, it follows that the subset of edges for each connected component will likewise be ordered by cost and that an MCST will be generated for each connected component since the entire set of edges is sorted by cost.

On the other hand, Prim's algorithm **won't produce a spanning forest**. The technique will technically fail if it is applied to a disconnected graph. This is the situation because Prim's algorithm will never be able to reach other related components because it starts by picking a random vertex and then builds by adding edges that can be reached from that vertex. There will always be a circumstance when an edge with one vertex in B and one not in B does not exist, which will inevitably produce a problem in line 4. The loop will never end since B will never equal V. The specifics of what will transpire depend on the implementation, but B will be an MCST for a single, arbitrarily connected component at the moment of failure.

2. **Adapt Prim's algorithm to graphs that may include edges of negative costs; give an example of an application where negative costs may occur naturally.**

Ans. Naturally, both negative and positive weights are handled by Prim's algorithm without any problems. One method to demonstrate this is to make all of the weights positive before processing by adding an arbitrarily big constant, and then in the final MCST, deduct this from the edges. It is clear that the same edges will be chosen and the same MCST will be produced whether or not this constant is included. Taxi drivers are one instance of a situation where negative weights naturally exist. Driving to a passenger might be seen as a negative cost, and driving a passenger to a location can be seen as a positive cost.

3. **A binary heap data structure is an array that we can view naturally as a nearly complete binary tree. Each node of the tree corresponds to an element in the array, as shown below:**



16	14	10	8	7	9	3	2	4	1

note that the array has the following interesting structure: the parent of i is bi/2c and the left child of i is 2i and the right child of i is 2i + 1. With the binary heap data structure we can implement line 4. efficiently, that is, find the cheapest e. To this end, implement a min-priority queue B, which, during the execution of Prim's algorithm, keeps track of all the vertices that are not in the tree T. The min-priority queue B uses the following key attribute: minimum weight of any edge connecting the vertex to T (and ∞ if no such edge exists). Describe the details of the above scheme, and implement it in Python 3.

Ans. Our Solution for Prims Algorithms can be found in the file Algorithms_Assignment_2_team_3.py or in the below link

https://drive.google.com/file/d/1FBhAhdU7oI9GDFBbT7dtaMDRJaBjRSXM/view?usp=sharing