**4. Difficulties Faced during the Phase Design**

The whole project was a bit hard for us because it using object-oriented programming but despite all of that we have found the right solution and output. Here we are going to list some difficulties we faced in some methods.

▪ Linking the two packages Graphframwork and PhonNetworkDesign was a bit hard because we could not imagine how to connect theses tow packages together.

▪ **makeGraph**: makeGraph doesn't clearly indicate what is incorrect given the large number of vertices and edges.

▪ **addEdge:** At first, it was challenging to get the logic idea of adding edge without having an array of vertices in parameter.

▪ **displayResultingMST for MHPrimAlg:** the problem was during the design we didn’t fully understand MHPrim apparently, because MHPrim method kept giving incorrect result of cost even though the weight is correct.

▪ **displayResultingMST for KruskalAlg:** Since we had to select the implementation of a disjoint subset, getting to the final design was difficult. Following the lists concepts for each representative with different head containing first/last/size was a bit difficult. so instead of that we dealt fully with the representative subsets array only in detailed conditions statements.

▪ **Office/Line**: It was kind of hard to get the information of these classes, but we found it easy to get it through Vertex/Edge Classes by using MSTresultList.

**5 Conclusion**

To conclude, the empirical analysis has succeeded in reaching a prospective answer between these two algorithms. We have compared between Kruskal’s algorithm with Quick find implementation and Prim’s algorithm with Min-Heap, it was clear that the min heap-based Prim’s algorithm is more efficient than Kruskal’s algorithm in finding the minimum spanning tree. Since Prim algorithm work on connected edges approach without addition implementation as Kruskal algorithm to make set and check each set to work its unconnected edges approach. Finally, we have successfully identified the right optimal algorithm of the minimum spanning tree problem.