**ASSIGNMENT-3**

* The code is used to generate MST using prim’s and kruskal’s algorithm on any given input graph and prints the respective MSTs , total cost and the execution time of the algorithm.
* To run the code we need to run the find.java file and we will get the desired output.
* The code consists of following 6 classes:-

1. **graph.java-** It is an implementation of a data structure which implements the GRAPH ADT. It has following methods:-

* **graph()-** A constructor to initialize the graph.
* **addEdge()-** It add a new edge in the graph.
* **display()-** It displays the graph.

1. **Prim.java-** It is an implementation of Prim’s algorithm, as a function which takes a GRAPH instance as input and returns another GRAPH instance as output. It has following methods:-

* **getPrim()-** Method that takes instance of graph and implements prim’s algorithm and returns another instance of graph.

1. **priority\_q.java-** It is an implementation of a data structure which implements the PRIORITYQUEUE ADT. It contains following methods:-

* **priority\_q()-** Constructor to initialize the priority queue.
* **delete\_min()-** Returns minimum value from the queue.
* **heapify()-** It restores the heap property on the node on which it is called.
* **insert()-** To insert a new element in the priority queue.
* **decrease\_key()-** To decrease the key value of any particular node.
* **display()-** To display the priority queue.

1. **mfset.java-** It is an implementation of a data structure which implements the MFSET ADT. It contains following methods:-

* **mfset()-** To initialize the mfset.
* **merge()-** To merge two sets of mfset together.
* **find()-** To find the set of any given element and return it.
* **initial()-** To add any new element in a set.
* **display()-** To display the mfset.

1. **kruskal.java-** It is an implementation of Kruskal’s algorithm. It has following methods:-

* **getKruskal()-** An implementation of Kruskal’s algorithm, as a function which uses the above two data structures, and takes a GRAPH instance as input and returns another GRAPH instance as output.

1. **find.java-** The top-level class which contains the main method. It has following methods:-

* **main()-** A function that takes as input the cost matrix for a graph in the given format, creates a GRAPH instance using it, calls the Prim and Kruskal functions on this instance and gets the GRAPH instances returned by each of them (whilst also keeping track of the runtime for each call), and prints out these MSTs in the given format along with the respective total costs and runtimes.