# How To Execute Long Project 2

Please find steps to execute Long project 2 below.

For compiling, extract the zip file to a folder and execute following command from extracted folder.

1. javac *.java
2. For Prim 1 execution, execute following command, please provide input path to the undirected Graph data as shown below.

```
G99_LP2>java MSTUsingPriorityQueue
"S:\IMPL_DS\Algorithms\MinimumSpanningTree\src\lp2-data\0-lp2.txt"
Statistics for Step Minimum Spanning Tree Function :
Time: 1 msec.
Memory: 3 MB / 192 MB.
---------------------------------------
Weight of the tree found to be 12
```

3. For Prim 2 Execution, execute following command, please provide input path of the undirected graph file.

```
G99_LP2>java MSTUsingIndexedPriorityQueue
"S:\IMPL_DS\Algorithms\MinimumSpanningTree\src\lp2-data\0-lp2.txt"
Statistics for Step Minimum Spanning Tree Function :
Time: 1 msec.
Memory: 3 MB / 192 MB.
---------------------------------------
Weight of the tree found to be 12
```

4. For Kruskals Execution, execute following command, Please provide input path of the undirected graph input data file.

```
G99_LP2>java KruskalsAlgorithm
"S:\IMPL_DS\Algorithms\MinimumSpanningTree\src\lp2-data\0-lp2.txt"
Time: 0 msec.
Memory: 3 MB / 192 MB.
---------------------------------------
Total Weight of the MST was found to be : 12
```

5. For Edmond's algorithm execution, please execute following command. Please provide input path of the directed graph file

```
G99_LP2>java EdmondsAlgorithm
"S:\IMPL_DS\Algorithms\MinimumSpanningTree\src\lp2-data\0-lp2.txt"
Statistics for Step => input graph read statistics :
Time: 21 msec.
Memory: 3 MB / 192 MB.
----------------------------------------
Statistics for Step => Edmonds : Minimum Spanning Tree retrieval :
Time: 3 msec.
Memory: 3 MB / 192 MB.
----------------------------------------
Total Weight of the MST was found to be : 17
(5,2)
(4,3)
(1,4)
(3,5)
```