Project 3

Problem 1:

Approach:

Merge.java: Merge class contains function to merge graph G1 and G2. Using arraylist for nodes, distance and coordinates for each graph then combining both graphs on basis of that no node has duplicate coordinates. (compile Merge.java first to get .class)

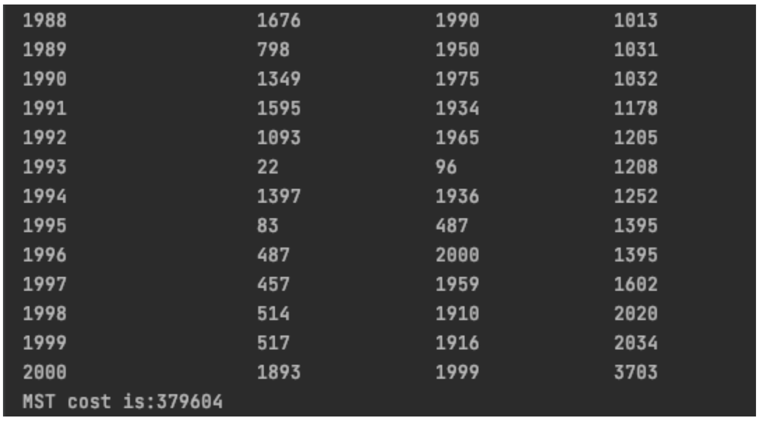
kruskal2D.java: Call function in merge.java to get G graph matrix and executes kruskals algorithm *(takes little time to run)*

kruskallinked.java: Call function in merge.java to get G graph matrix and then initializes graph class for linked list implementation. Only edges whose cost is not equal to Integer.MaxValue are added

(Note: To print memory usage uncomment line 68,69 in kruskal2D.java and line 124,125 in kruskallinked.java)

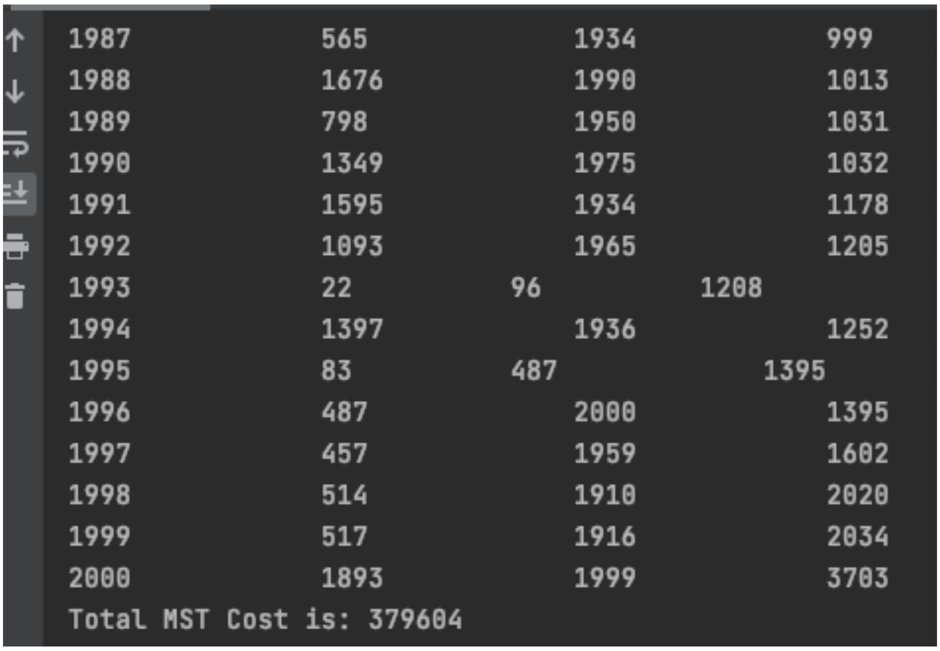
MST tree is printed in the form EdgeNumber(starts from 0) Node Node Distance

Kruskal2D



```
1988            1676        1990        1013
1989            798         1950        1031
1990            1349        1975        1032
1991            1595        1934        1178
1992            1093        1965        1205
1993            22          96          1208
1994            1397        1936        1252
1995            83          487         1395
1996            487         2000        1395
1997            457         1959        1602
1998            514         1910        2020
1999            517         1916        2034
2000            1893        1999        3703
MST cost is:379604
```

Kruskal Linked:



```
1987            565             1934            999
1988            1676            1990            1013
1989            798             1950            1031
1990            1349            1975            1032
1991            1595            1934            1178
1992            1093            1965            1205
1993            22          96          1208
1994            1397            1936            1252
1995            83          487             1395
1996            487             2000            1395
1997            457             1959            1602
1998            514             1910            2020
1999            517             1916            2034
2000            1893            1999            3703
Total MST Cost is: 379604
```

Memory Usage

With adjacency matrix the memory used is around 1.72% of my total memory whereas the memory used with adjacency list is around 1.31%. Memory with linked list is lower because it does not store empty or 0 edges so this saves up the overall memory. With matrix we have empty edges stored which is unnecessary space utilized. Execution with linked list is comparatively faster than matrix. Looking at the percentages it won't look like a much significant difference between the two memory because the graph is quite dense i.e. the number of edges is close or more to the maximal number of edges.