***Graphs MST***

***Name: Sherif Mohi Mahmoud Hafex***

***Student number: 27***

**Objective:**

Implementing an algorithm to find minimum spanning tree and analyzing it.

**Requirements**

You are required to write procedures (or subprograms) to implement finding the minimal cost spanning tree using both Prim’s and Kruskal's algorithms. You should design and implement appropriate tests for testing it, and provide appropriate analysis for the complexity of both algorithms.

**Suggestions for Testing:**

1- Compare different Graph representations (Adjacency list and Adjacency Matrix).

2-Compare different Graph sizes and densities (dense and sparse graphs with different sizes).

3Testing against randomly generated graphs.

***Testing:***

***I made three methods to generate graphs:***

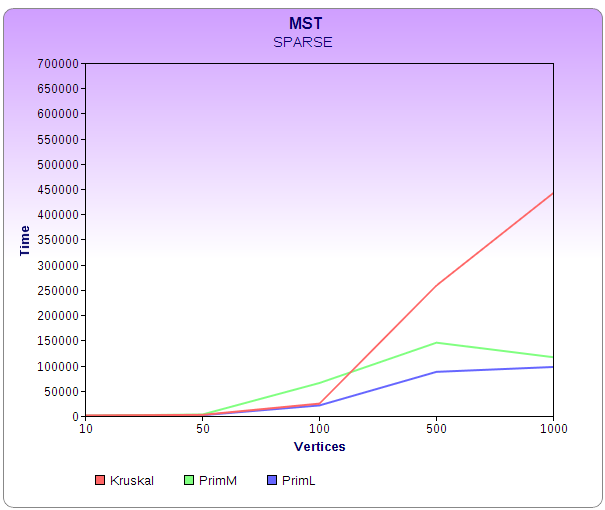
1- Dense: to make a dense connected graph with any number of vertices.

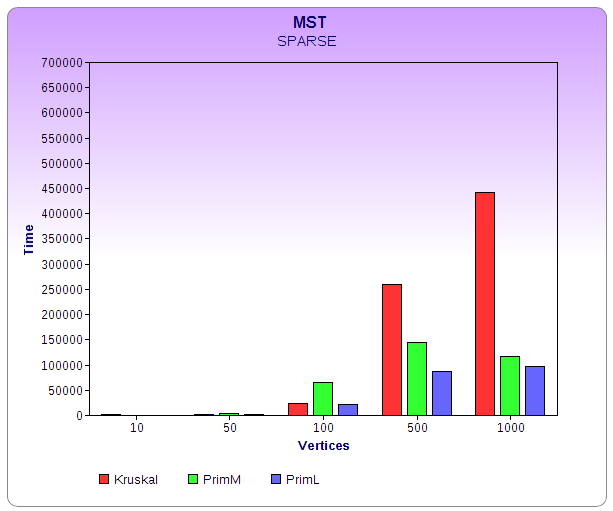
2- sparse: to make a dense connected graph with any number of vertices.

3- Random: to generate any connected graph.

***Note :*** kruskal algorithm has one implementation using Edge class we don't need to make the graph representation using adjacency matrix or adjacency list so, ***the test will be between kruskal algorithm, prim algorithm using adjacency Matrix, prim algorithm using adjacency list.***

***Test sparse graph using different number of vertices (different size):***

******

******

In most cases we found that the prim algorithm using adjacency list is better than kruskal and prim using adjacency matrix algorithms.

***Examples using sparse method:***

Vertices: 10 edges (11)

Kruskal: 1145 Microsecond primM : 666 M.S primL: 152M.S

This case indicates that primL is better than primM that isn't useful in case of most sparse graphs. It is also better than kruskal because union find's time has an important effect.

Vertices: 50 edges (265)

Kruskal: 2193 Microsecond primM: 2665 M.S primL: 1956M.S

This case indicates primL Is better than kruskal but kruskal is better than primM because the number of vertices increased so, the size of the matrix increased so it will take more time in this case.

Vertices: 50 edges (101)

Kruskal: 2228 Microsecond primM: 3290 M.S primL: 1157M.S

Vertices: 100 edges (2247)

Kruskal: 39250 Microsecond primM : 29985 M.S primL: 30641M.S.

This case indicate primM is better than kruskal and primL

Vertices: 100 edges (2190)

Kruskal: 9853 Microsecond primM : 14317 M.S primL: 11351 M.S.

This case indicate kruskal is better than primM and primL.

Vertices: 500 edges (45721)

Kruskal: 258819 Microsecond primM : 145391 M.S primL: 87557 M.S.

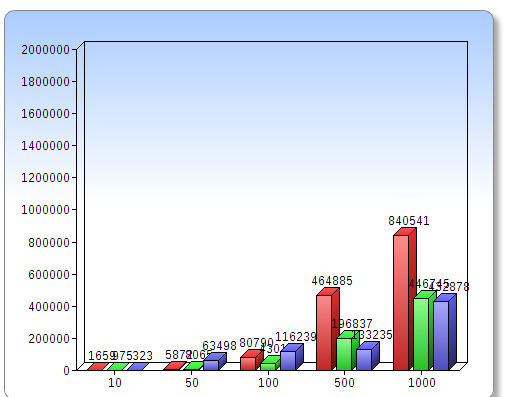
This case indicate primL is better than kruskal and prim

Vertices: 1000 edges (87009)

Kruskal: 442382 Microsecond primM : 116556 M.S primL: 97190 M.S.

This case indicate primL is better than kruskal and prim.

***Test Dense graph using different number of vertices (different size):***

******

Vertices: 10 edges (32)

Kruskal: 1659 Microsecond primM : 975 M.S primL: 323 M.S.

primL is better than kruskal and prim.

Vertices: 50 edges (696)

Kruskal: 5146 Microsecond primM : 8045 M.S primL: 6947 M.S.

Kruskal is better than primL and primM.

Vertices: 50 edges (1210)

Kruskal: 6598 Microsecond primM : 8086 M.S primL: 10757 M.S.

Kruskal is better than primL and primM.

Vertices: 100 edges (4709)

Kruskal: 80790 Microsecond primM : 43014 M.S primL: 116239 M.S.

PrimM is better than kruskal and primL

Vertices: 500 edges (104842)

Kruskal: 464885 Microsecond primM : 196837 M.S primL: 133235 M.S.

primL is better than kruskal and prim

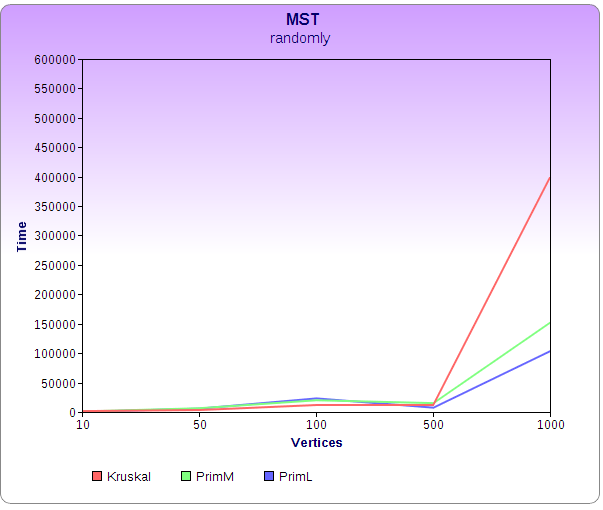
Vertices: 1000 edges (32848)

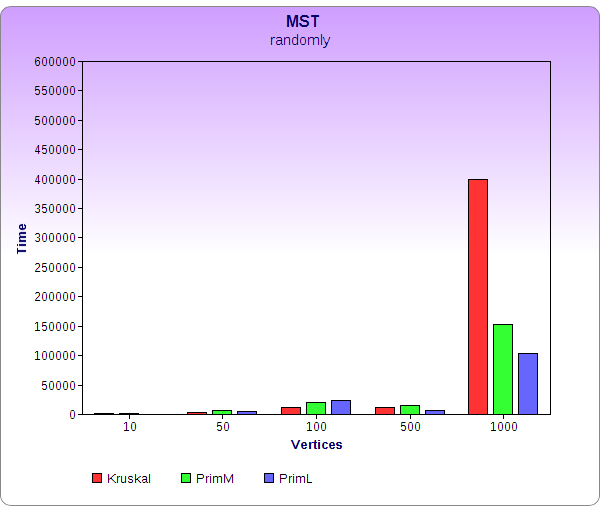
Kruskal: 840541 Microsecond primM : 446745 M.S primL: 432878 M.S.

primL is better than kruskal and primM.

When number of vertices and edges increases, primL tends to be the best.

***Test Random graph using different number of vertices (different size):***

******

******

Vertices: 10 edges (16)

Kruskal: 1525 Microsecond primM : 875 M.S primL: 197 M.S.

primL is better than kruskal and primM.

Vertices: 50 edges (727)

Kruskal: 3341 Microsecond primM : 6399 M.S primL: 5748 M.S.

kruskal is better than primL and primM.

Vertices: 100 edges (2091)

Kruskal: 11856 Microsecond primM : 19937 M.S primL: 23087 M.S.

kruskal is better than primL and primM.

Vertices: 100 edges (1314)

Kruskal: 11903 Microsecond primM : 15065 M.S primL: 7365 M.S.

primL is better than kruskal and prim.

Vertices: 500 edges (78511)

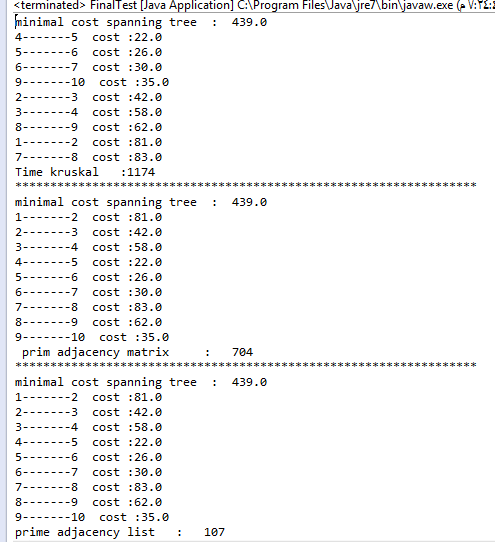
Kruskal: 399658 Microsecond primM : 152261 M.S primL: 103631 M.S.

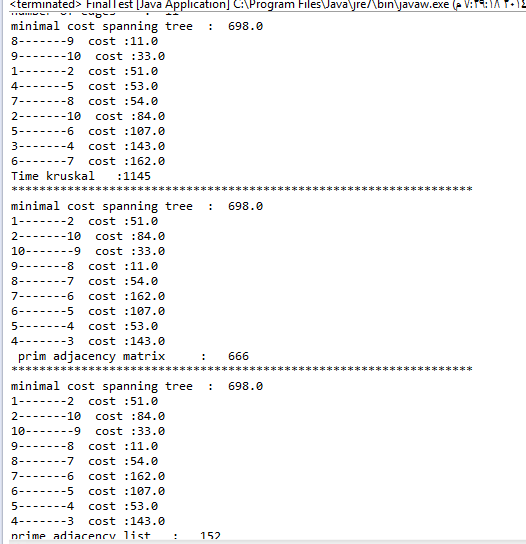
primL is better than kruskal and prim.

Vertices: 1000 edges (7453)

Kruskal: 127329 Microsecond primM : 84768 M.S primL: 31066 M.S.

primL is better than kruskal and prim.

******

******