Minimum Spanning Tree

Minimum spanning tree (MST) is a fundamental research problem with various real-life applications. Please see the attached file "mst.pdf" for more detailed introductions and solution algorithms. Here are the requirements for our homework:

(1) Implement a class named MST in header file mst.h and source file mst.cpp.

(2) Implement a class named Graph in header file graph.h and source file graph.cpp to store the graph with positive edge weights. Each node in the graph represents a point with 2D coordinates (x,y). The edge weight is a float variable that denotes the Euclidean distance between the corresponding nodes.

(3) Implement a public member function "computeMST" in class MST to compute the minimum spanning tree. Either Kruskal's or Prim's algorithm is ok.

(4) In main.cpp file, randomly generate 20 distinct points (x,y) in the 2D plane (0 <= x,y <= 10000). No overlap is allowed between the 20 points. Construct the object g of class Graph from the given points. Compute the minimum spanning tree of g by the public interface "computeMST".

(5*) [**This is not mandatory to finish, but it is a challenging topic**] Can you provide a public interface "computeTopKMST" compute the top K (1 <= K <= 20) minimum spanning trees of g?

Tips:

1. You are free to design your classes, e.g., you may add any new classes, any new member functions and member variables to existing classes, etc.