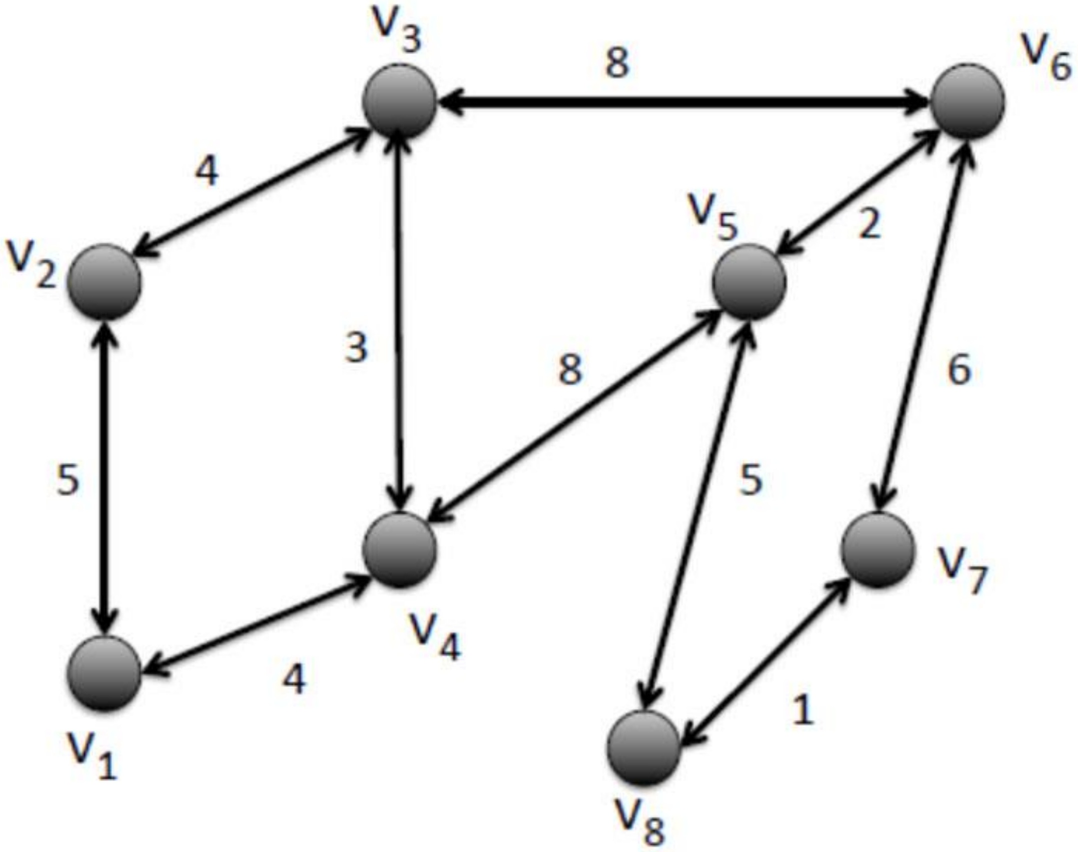
1. Conduct a small research on the minimum spanning tree problem and efficient algorithms to solve it. You can find all required details in chapter 14.7 of the course book “Data Structures and Algorithms in Java”. Learn how to solve the problem via the two particular solution techniques, Prim-Jarnik’s and Kruskal’s algorithms, described, respectively, in sections 14.7.1 and 14.7.2 of the book. You may of course explore and refer to any other resources covering this topic. We expect you to grasp the idea and important facts about the problem as well as the runtime complexity, implementation issues, and advantage(s) of each of the two algorithms. As the result of your study, you must be able to explain the both algorithms and how they work.

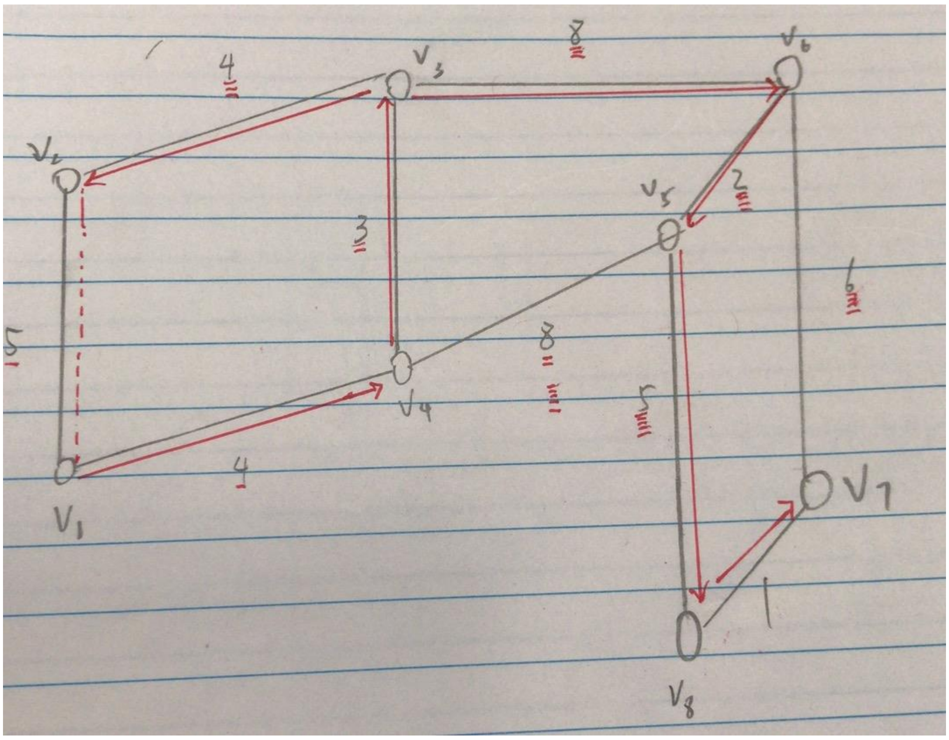
Runtime = O(nlogn)

minimum spanning tree problem have two algorithm:

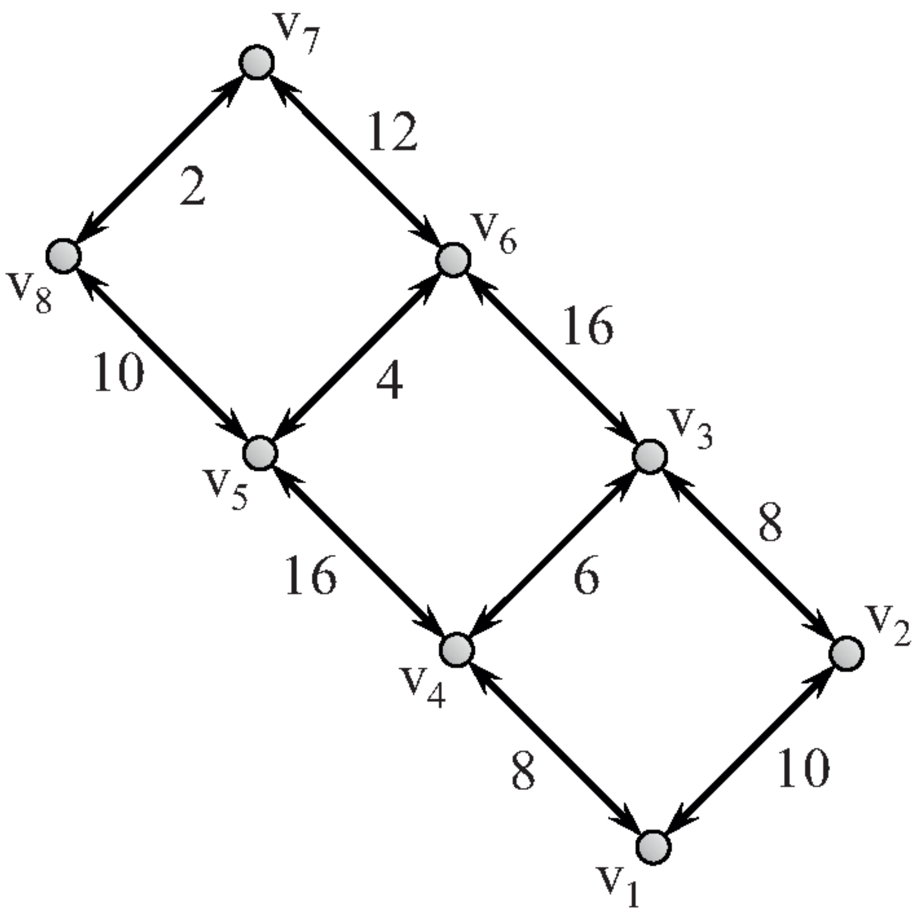
1. Prim algorithm: this logic is use the Dijkstra、Bellman-Ford algorithm
2. Kruskal algorithm: this one is use the sorting the each edge’ s length form small to large. Link all small edge to create new tree.

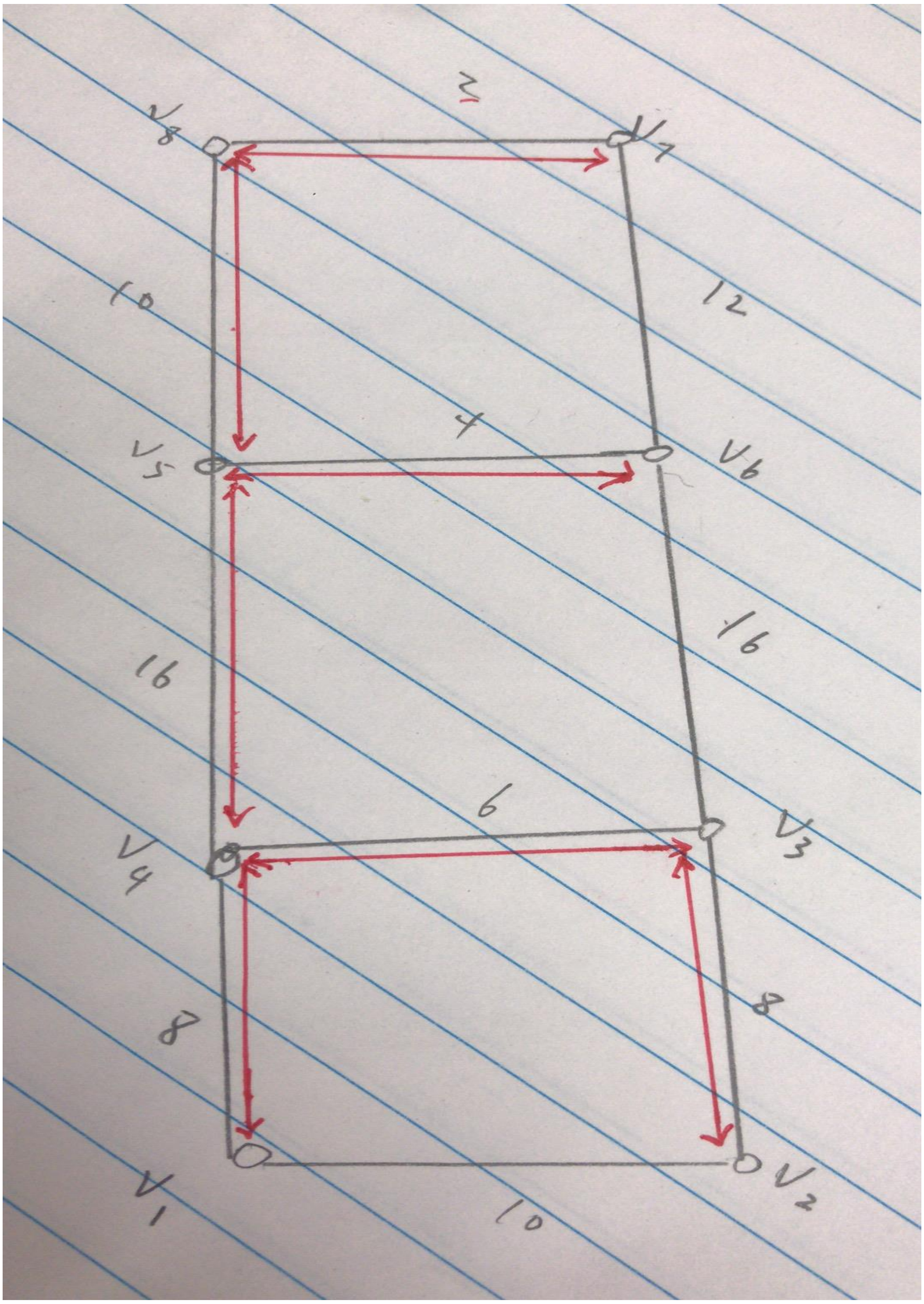
2. Solve the following numeric example. For the given undirected (bi-directed) graph, compute a minimum spanning tree using Prim-Jarnik’s algorithm. Show the structure of your partial minimum spanning tree after each edge insertion and indicate for each edge whether it is included in the minimum spanning tree.





3. Solve the following numeric example. For the given undirected (bi-directed) graph, compute a minimum spanning tree using Kruskal’s algorithm. Show the structure of your partial minimum spanning tree after each edge insertion and indicate for each edge whether it is included in the minimum spanning tree.


v_7
12
2
v_8
v_6
16
10
4
v_3
v_5
8
16
6
v_2
v_4
8
10
v_1


2
V8
V7
10
12
4
V5
V6
16
16
6
V4
V3
8
8
V1
V2
10