Algorithms: Assignment #7

Tanvi Magdum
magdum.t@northeastern.edu

February 27, 2023

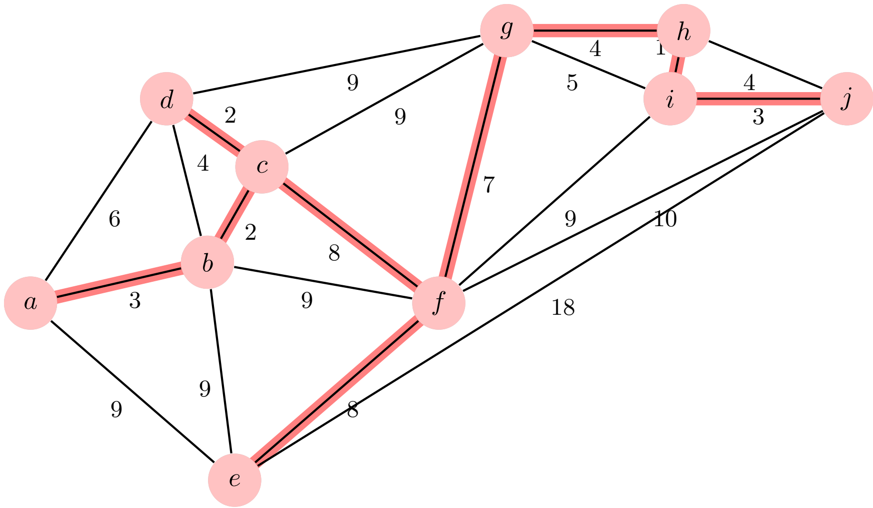
Answers

1 Problem 1 Solution :

Prim's algorithm is a greedy algorithm used to find the minimum spanning tree (MST) of a weighted undirected graph. MST is a tree that connects all the vertices in the graph with minimum possible total edge weight.

For given problem, the algorithm can be designed as follows -

1. We have a priority queue PQ = {A, B, C, D, E, F, G, H, I, J}. Since, we are starting from vertex A, add it to the MST. So, MST = {A} and PQ = {B, C, D, E, F, G, H, I, J}. Keep adding new vertices to MST until PQ is not empty.
2. For each adjacent vertex of A, calculate the weight of edge connecting A and the adjacent vertex. We have B, D, E as adjacent vertices of A with weights AB = 4, AD = 6, AE = 9.
3. Choose the vertex with smallest edge weight and add it to MST. So, MST = {A, B} and PQ = {C, D, E, F, G, H, I, J}.
4. Now, consider adjacent vertices of A and B both since both are a part of MST now. Those are C, D, E, F.
5. Then find the edge with minimum weight connected to MST, that is, vertices A or B or both. We have AD = 6, AE = 9, BC = 2, BD = 4, BE = 9 and BF = 9. The minimum weighted edge would be BC, so add vertex C in MST. Hence, MST = {A, B, C} and PQ = {D, E, F, G, H, I, J}.
6. Repeat the same process for each adjacent vertex to the vertices present in MST till now. Now, we have adjacent vertices as D, E, F, G.
7. Edge weights are AD = 6, AE = 9, BD = 4, BE = 9, BF = 9, CD = 2, CF = 8, CG = 9. We discard edges AD and BD because they will form a cycle in the graph. Anyway, the minimum weighted edge is CD. So, MST = {A, B, C, D} and PQ = {E, F, G, H, I, J}.
8. Now outer tree vertices connected to MST are E, F, G.
9. AE = 9, BE = 9, BF = 9, CF = 8, CG = 9, DG = 9. We excluded edges forming cycles. Thus, shortest edge is CF so MST = {A, B, C, D, F} and PQ = {E, G, H, I, J}.
10. The outer tree vertices connected to MST are E, G, I, J.
11. Edges AD, BD, BF form cycle so exclude them. Other edges, AE = 9, BE = 9, CG = 9, DG = 9, FE = 8, FG = 7, FI = 8, FJ = 10. Smallest edge is FG, so add G in MST. MST = {A, B, C, D, F, G} and PQ = {E, H, I, J}.
12. Now, outer tree vertices connected to MST are E, H, I, J.
13. Excluding edges forming cycles, other edges are AE = 9, BE = 9, FE = 8, FI = 8, FJ = 10, GH = 4, GI = 5. The minimum distance is GH. So, MST = {A, B, C, D, F, G, H} and PQ = {E, I, J}.



14. Outer tree vertices connected to MST are E, I, J.

15. Excluding edges forming cycles, other edges are AE = 9, BE = 9, FE = 8, FI = 8, FJ = 10, GI = 5, HI = 1, HJ = 4. The smallest distance is HI, so MST = {A, B, C, D, F, G, H, I} and PQ = {E, J}.

16. Now remaining outer tree vertices are E and J that are connected to MST.

17. Discarding all edges that form cycles, AE = 9, BE = 9, FE = 8, FJ = 10, HJ = 4, IJ = 3. IJ is smallest so J becomes part of MST. MST = {A, B, C, D, F, G, H, I, J} and PQ = {E}.

18. Only vertex E remains with edges AE = 9, BE = 9, FE = 8, JE = 18. Smallest distance is FE. So, MST = {A, B, C, D, F, G, H, I, J, E} and PQ is now empty. So we can stop the algorithm now.

We have 10 nodes and 9 edges in this newly created MST. The tree would look something like above (in red).

.

The sum of the edge weights in this MST would be -

3+2+2+8+7+4+1+3+8 = 38. Therefore, the sum of the weights of the edges in the MST is 38.

2 Problem 2 Solution :

If G is a dense graph with extremely large number of vertices, then Kruskal's algorithm will output MST faster than Prim's algorithm. Kruskal's algorithm sorts edges in the graph by its weight and then iterates over them. It selects edges having lowest weights that do not form cycle. It iterates until all the vertices in given graph are included in MST. This algorithm has a runtime complexity of O(E log V), where E is the number of edges in the graph and V is the number of vertices.

Prim's algorithm starts by selecting an arbitrary vertex and grows MST by selecting an edge with the lowest weight that connects a vertex in the minimum spanning tree to a vertex outside this tree. Prim's algorithm has a time complexity of O(E + V log V).

In a dense graph with large number of vertices, edges are most likely to be larger in amount than the vertices in a graph. Maximum number of edges a graph can have is $V * (V - 1)/2$. Therefore, in a dense graph, number of edges E is proportional to V^2. Comparing the time complexity of both algorithms,

Kruskal's : $O(ElogV) = O(V^2logV)$

Prim's : $O(E + VlogV) = O(V^2 + VlogV)$

For large values of E, term V^2 dominates both time complexities. So, we need to compare the logarithmic terms. This will indicate how running time increases with respect to the size of the input. In Kruskal's, running time increases logarithmically with V, and in Prim's it increases linearly with V. Hence, (E log V) term in Kruskal's algorithm dominates (E + V log V) term in Prim's, if value of V is large enough.

Thus, Kruskal's Algorithm would output the minimum weight spanning tree more quickly than Prim's Algorithm in a dense graph with an extremely large number of vertices.

3 Problem 3 Solution :

3.1 Problem 3.1 Solution :

The given graph can with 8 vertices can form a large number of edges, since we can assume that every node is connected to every other node. So, this is a dense graph. It is wise to use Kruskal's algorithm on such graph. It is one of the algorithms to find the minimum spanning tree of a connected weighted graph. It begins by sorting the edges by weight and then adding next smallest edge. This edge should not form a cycle in graph with the edges that are already present in MST.

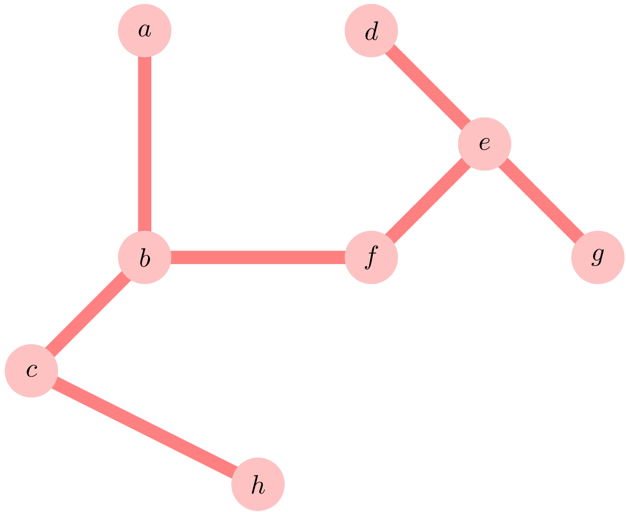
Assume that distance between two adjacent grid points is 1.

1. Sort the edges in graph by weight and store them.
2. Since, all the vertices are present at the grid points, we know that all points are proportionately apart from each other. The minimum distance between points could be 1 according to the assumption. But in given graph, we don't have any points with 1 unit distance apart. The minimum distance between two points in this graph is $\sqrt{2}$. Edges bc, de, eg and ef have $\sqrt{2}$ distance.
3. Choose any edge one edge and add both vertices to MST. Edges = {b-c} and MST = {b, c}.
4. Now add another edge with weight $\sqrt{2}$, edge de. Edges = {b-c, d-e} and MST = {b, c, d, e}.
5. Now add another edge with weight $\sqrt{2}$, edge ef. Edges = {b-c, d-e, e-f} and MST = {b, c, d, e, f}.
6. Now add last edge with weight $\sqrt{2}$, edge ef. Edges = {b-c, d-e, e-f, e-g} and MST = {b, c, d, e, f, g}.
7. Until now, no edge has formed any cycle in graph. Add next smallest edge now. It would be edge with weight 2. There are 5 such edges, ab, ad, bf, df, fg.
8. We exclude edges df and fg because it forms a cycle. Now choose edge either of the 3 edges ab, ad or bf. Choose ab. Edges = {b-c, d-e, e-f, e-g, a-b} and MST = {b, c, d, e, f, g, a}.
9. Now if we select any of the edges ad or bf, no cycle is formed. Select bf. Edges = {b-c, d-e, e-f, e-g, a-b, b-f} and MST = {b, c, d, e, f, g, a}.
10. If we select ad, then cycle will be formed in graph. So, discard ad.
11. Now only one vertex is left - h. Next smallest edge is with distance $\sqrt{5}$. Edges hb, hc, hf have distance $\sqrt{5}$.
12. Choose any one edge, suppose hc. Edges = {b-c, d-e, e-f, e-g, a-b, b-f, h-c} and MST = {b, c, d, e, f, g, a, h}.
13. Now we have all vertices covered and total edges formed in MST are 7 for 8 vertices. So, we don't need to go other sorted edges. We can end the algorithm.

The total sum of edge weights in this MST would be -

$2 + 2 + \sqrt{2} + \sqrt{2} + \sqrt{2} + \sqrt{2} + \sqrt{5} = 4 + 4\sqrt{2} + \sqrt{5} = 4 + 5.65 + 2.24 = 11.89$ units

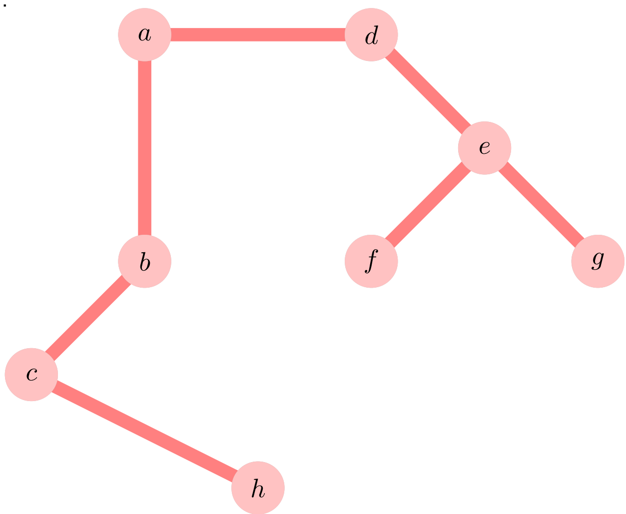
.



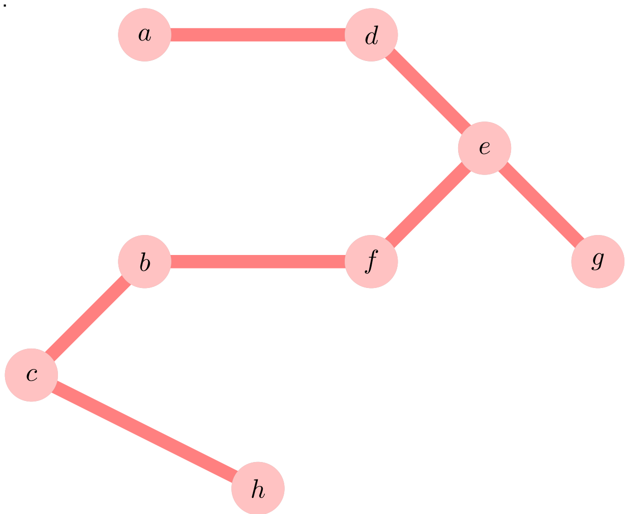
3.2 Problem 3.2 Solution :

Due to identical distances, there are many possible MSTs forming from the given graph. When we used Kruskal's algorithm, we saw there were many steps where we had options to choose edges. Below are the possible MSTs other than the one formed in previous answer-

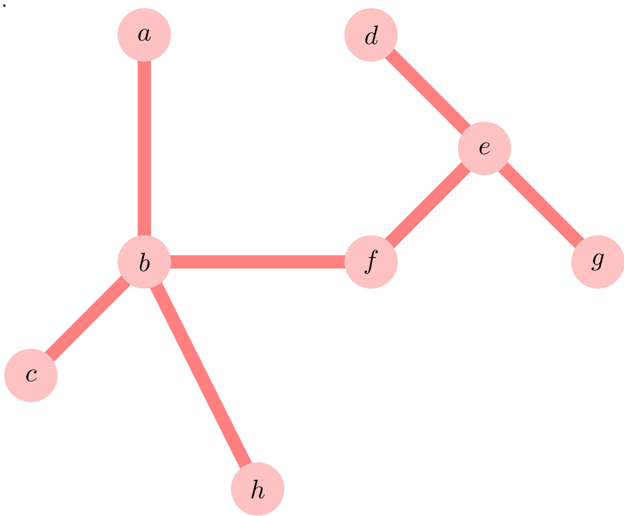
.



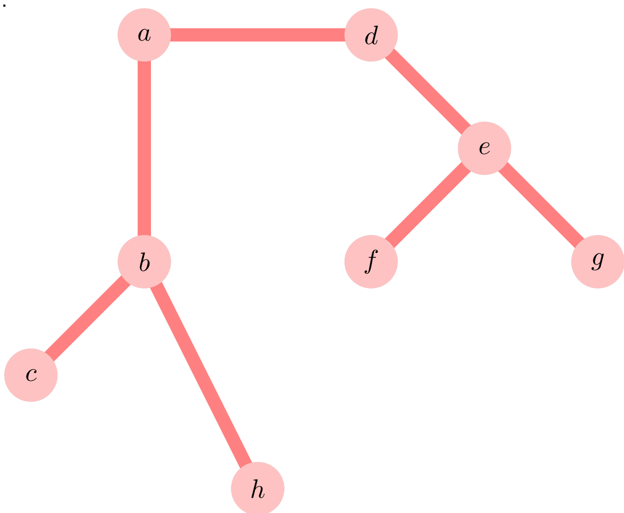
.



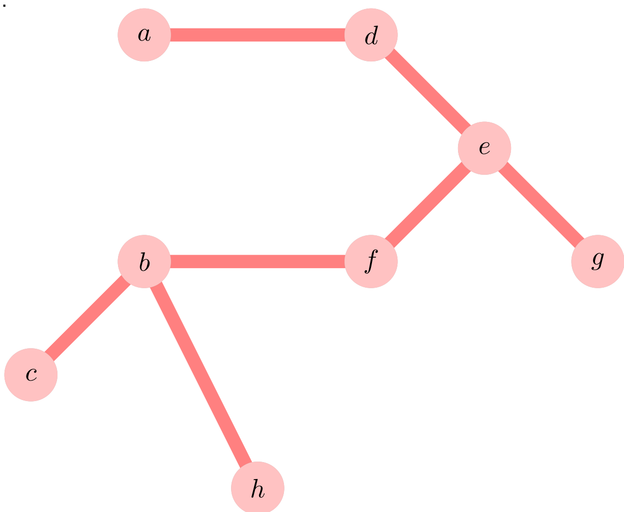
.


a
d
e
b
f
g
c
h

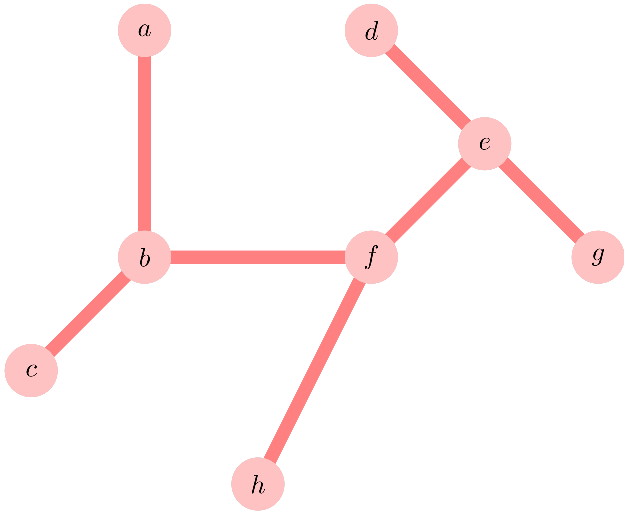
.


a
d
e
b
f
g
c
h

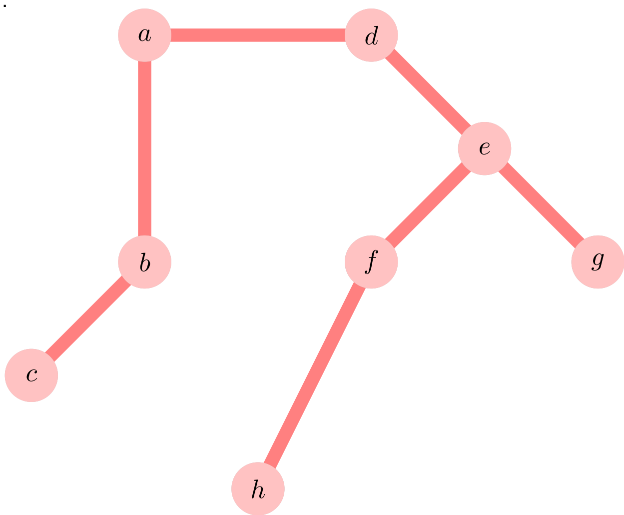
.


a
d
e
b
f
g
c
h

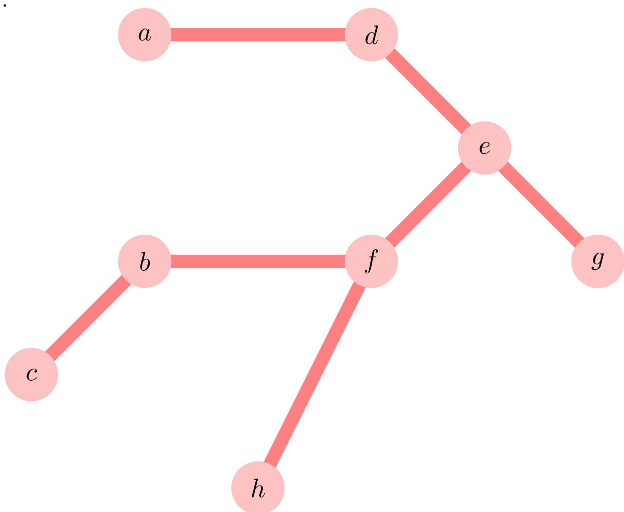
.



.



.

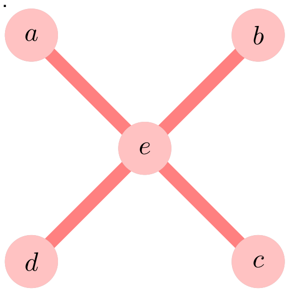


Hence, total 9 MSTs can be formed from this graph. The minimum distance here is $\sqrt{2}$ and we need to include all edges with this distance, until none is forming a cycle in graph. Since, we can include all 4 edges here, the inclusion of rest of the edges depend upon the MST created by edges with $\sqrt{2}$ distance. Hence, though the distance between edges fg and df is 2, we cannot include those because those edges to form any combination of MST as they form cycle. There was no possibility that we could have considered them first because, distance 2 is not smallest in our case. In short, we prioritize shortest weight edges first.

If we calculate sum of all edges for all of these graphs, it would come out to be same, as we have just replaced edges with same distance without adding or subtracting any edge weight. Thus, any graph can have multiple MSTs but the sum of edge weights for an MST would be constant for a particular graph, regardless of the algorithm used to calculate MST.

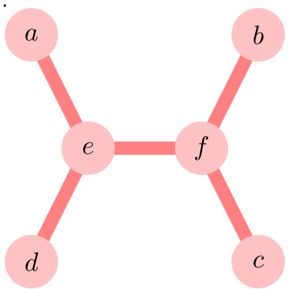
4 Problem 4 Solution :

In this problem, suppose we add a vertex E right at the centre of the given square. Then connect AE, BE, CE and DE.



Now let's calculate the sum of edge weights. Since, AB, BC, CD, AD all measure 100 making it a square, the length of diagonal will be $\sqrt{2}$ * side = $100\sqrt{2}$. So, AE, BE, CE and DE will be half of diagonal, that is $50\sqrt{2}$. Now adding all the distances, sum = 4 * $50\sqrt{2}$ = 282.84 units.

This distance is less than 300, so adding 1 vertex reduced the sum of edge distances and formed a better MST. Let us add 2 vertices and check if we get more superior MST. The vertices we are adding should be inside the square, then only the distance between 2 vertices will reduce. Let us add two vertice E and F in following fashion -



In this case, AE = DE = BF = CF. E is placed at 25 unit distance from midpoint of edge AD and F is at 25 unit distance from midpoint of edge BF. So, EF eventually turns out to be 50. To calculate DE, apply Pythagoras theorem. Drop a perpendicular from E on edge CD. Let the point be P. PE = 50 and PD = 25. So, DE = $\sqrt{50^2 + 25^2}$ = 55.9 $\approx$ 56.

Thus, sum of weights would be, sum = 4*56 + 50 = 274 units. Again as we added the new vertex, the MST got better. We got sum of weights reduced. Thus, 274 can be a minimum sum of edge weights when we add 2 extra vertices into given graph.