In Class 7

Tuesday, May 23, 2023 3:37 PM

CS2023 - Inclass Lab

Week 11 - MST

Note: You are required to answer the below questions and submit a PDF to the submission link provided under this week before the deadline (no extensions will be provided). You can either write / type your answers, but either way your answers should be readable.

Add the link to the GitHub repository

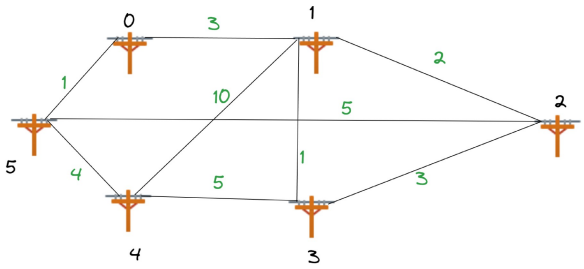
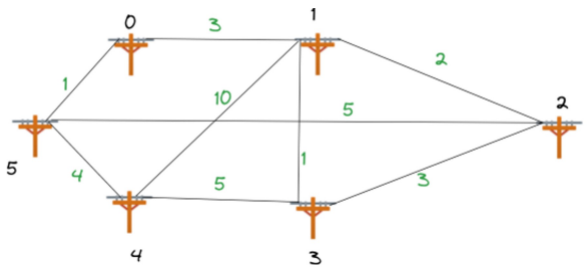


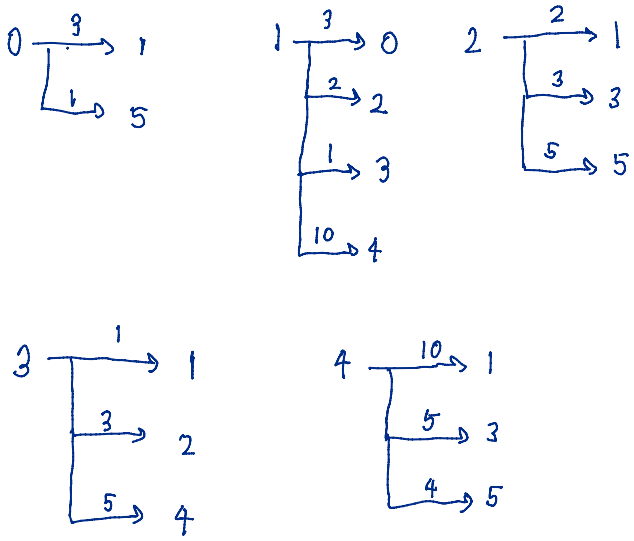
Figure 1: Telephone Pole Graph

Lab instruction

The problem is finding the minimum wire cost required by a telecommunication company to connect 6 streets. The linesmen have given you the possible wiring they can perform and the distance between each pole when the wiring is done. As an engineer, you must give them the best possible wiring connections between poles so that the wiring cost is minimal to your company. Expected submission

1. Write the adajaceny matrix for the graph on Fig 1.
2. Calculate and draw the minimum spanning tree for the graph in Fig1, taking **Node 3** as the start node.
3. Implement Prim's MST algorithm and obtain the minimum spanning tree taking **Node 0** as the start node. Take screen shot of your output.
4. Are the MSTs in Question 2 and Question 3 the same? What is the condition for a graph to have only 1 minimum spanning tree.
5. Discuss the time complexity between Prim's Algorithm and Kruskal's Algorithm.





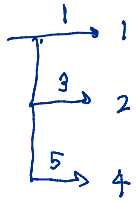
1. Write the adajaceny matrix for the graph on Fig 1.

$$\begin{pmatrix} 0 & 3 & 0 & 0 & 0 & 1 \\ 3 & 0 & 2 & 1 & 10 & 0 \\ 0 & 2 & 0 & 3 & 0 & 5 \\ 0 & 1 & 3 & 0 & 5 & 0 \\ 0 & 10 & 0 & 5 & 0 & 4 \\ 1 & 0 & 5 & 0 & 4 & 0 \end{pmatrix}$$

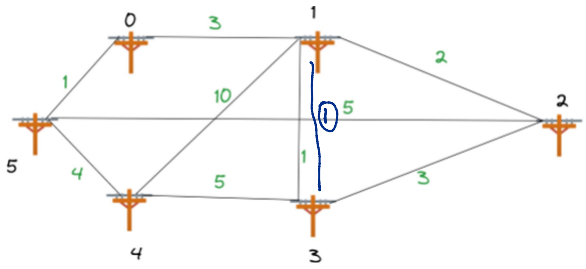
2. Calculate and draw the minimum spanning tree for the graph in Fig1, taking **Node 3** as the start node.

MST = [3] NON-MST = [0, 1, 2, 4, 5]

Possibilities

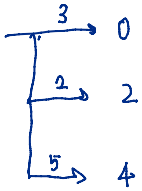


⇒ Next Step = 1

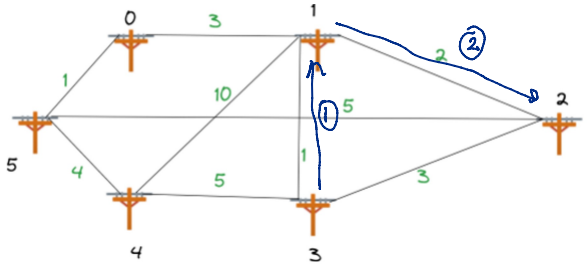


MST = [3, 1] NON-MST = [0, 2, 4, 5]

Possibilities

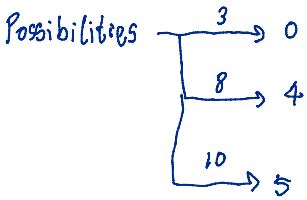


⇒ Next step = 2

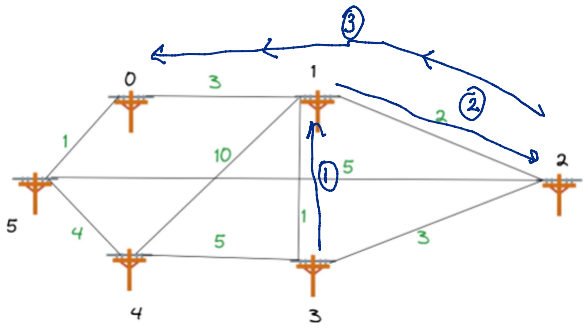


MST = [3, 1, 2] NON-MST = [0, 4, 5]

Possibilities



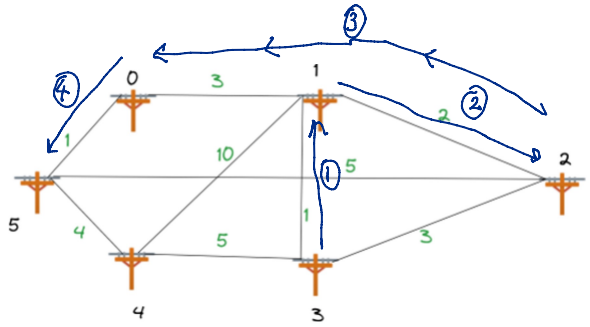
⇒ Next step = 0



MST = [3, 1, 2, 0] NON-MST = [4, 5]

Possibilities → 8 → 4
→ 1 → 5

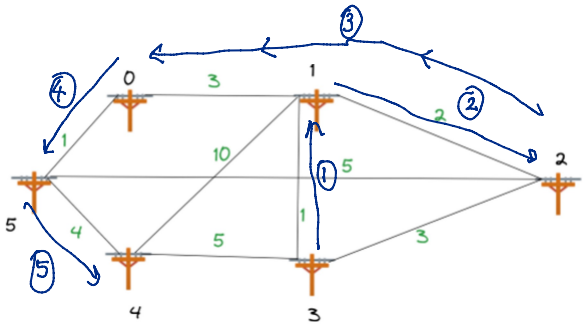
⇒ Next step = 5



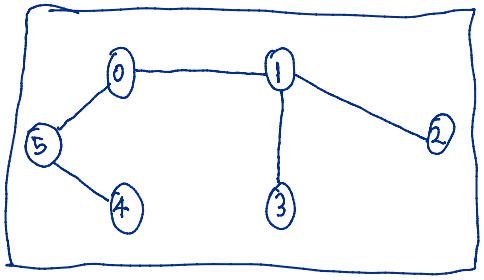
MST = [3, 1, 2, 0, 5] NON-MST = [4]

Possibilities → 4 → 4 | 4
→ 0 → 1 → 10 → 4 | 10
→ 0 → 1 → 2 → 3 → 3 → 5 → 4 | 8

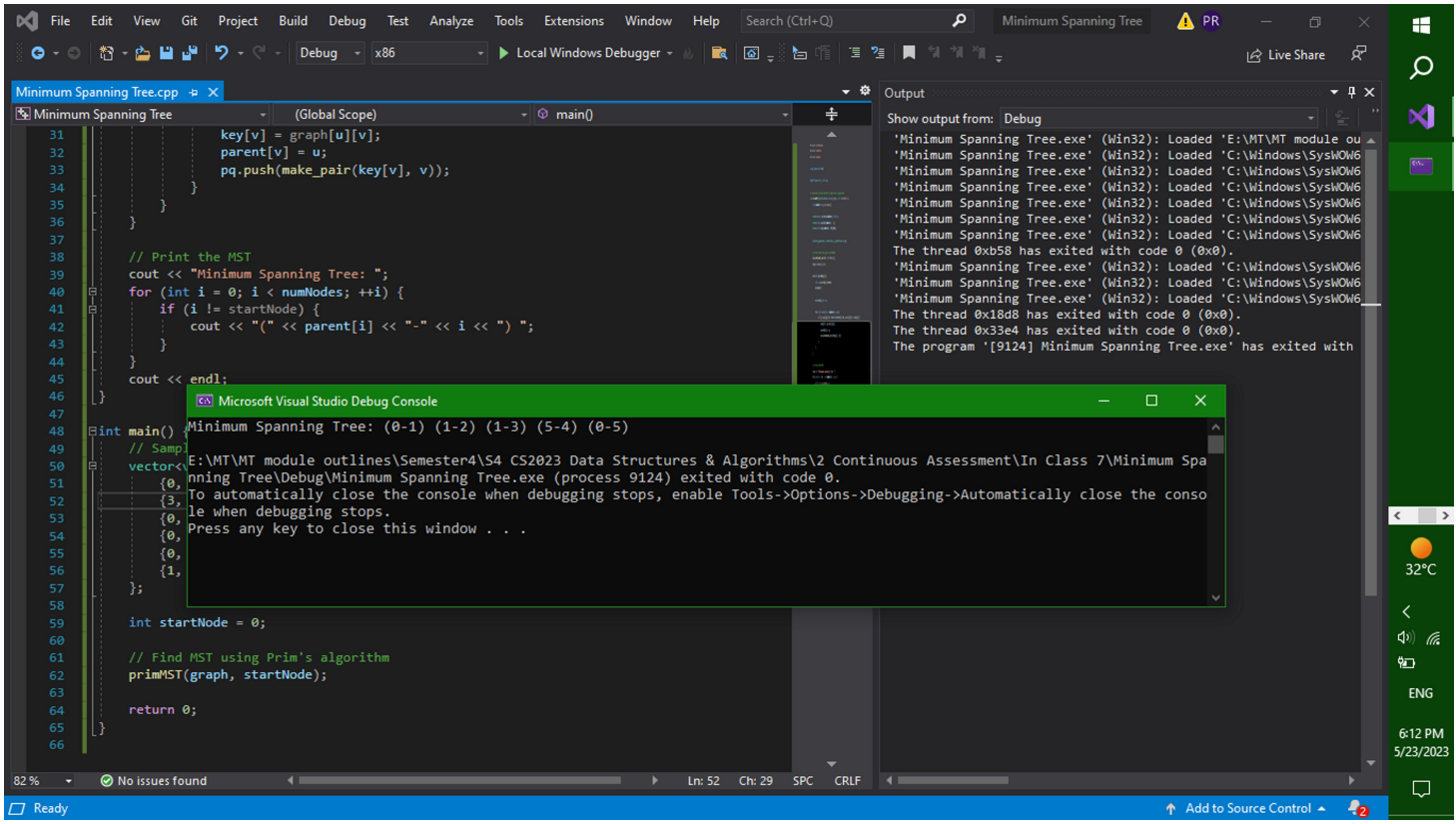
⇒ Next step = 4



MST = [3, 1, 2, 0, 5, 4]



3. Implement Prim's MST algorithm and obtain the minimum spanning tree taking **Node 0** as the start node. Take screen shot of your output.



4. Are the MSTs in Question 2 and Question 3 the same? Yes.

What is the condition for a graph to have only 1 minimum spanning tree.

If and only if, all the edge weights in the graph are unique, there will be only 1 minimum spanning tree.

5. Discuss the time complexity between Prim's Algorithm and Kruskal's Algorithm.

Time Complexity of,

Prim's Algorithm = $O(V^2)$

Kruskal's Algorithm = $O(E \log V)$

for E - Number of edges in the graph

V - Number of vertices in the graph.

In a situation where number of edges is larger compared to all possible edges that can be created by given vertices, Kruskal's Algorithm supports. In other conditions, Prim's algorithm is ideal.

Eg: 1) If number of edges is 990 & all possible edges that can be created by given vertices is 1000. Then Kruskal's algorithm supports than Prim's algorithm in time.

2) If number of edges is 600 & all possible edges that can be created by given vertices is 1000. Then Prim's algorithm supports than Kruskal's algorithm in time.

GitHub Link

https://github.com/200542P-MT-20/S4-CS2023-Data-Structures-Algorithms

The file is a Visual Studio file with .sln extension
Under In Class 7 (Week 11)