Optimization 2, Assignment 1
The Minimum Spanning Tree Problem

Group 1

Instructor: Prof. Nguyen Dinh

October 12, 2022

---

1 Group Members

- Tran Chau Thanh An - MAMAIU19001
- Tran Hoang Duy - MAMAIU19005
- Tran Viet Hang - MAMAIU18079
- Tran Thanh Hieu - MAMAIU19006
- Le Minh Hoang - MAMAIU19051

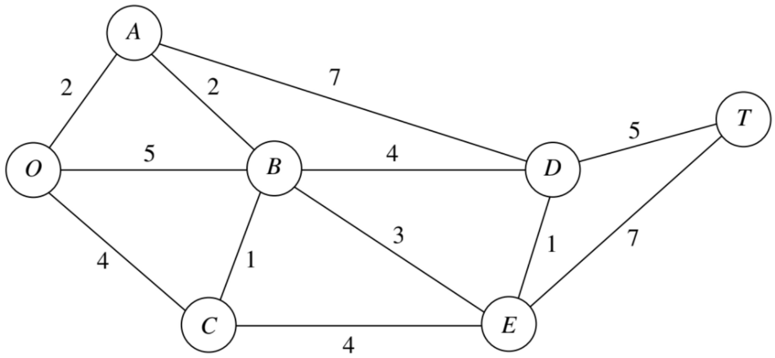
2 Abstract

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected network, whose arcs are undirected. This problem is to identify a shortest path to connect all the nodes in the network without creating any circles.

In this report, we will introduce Prim's algorithm and apply it to solve the minimum spanning tree problem given by our lecturer. The report is summarized with outlined notes about the problem.

3 Introductory Example

We now rephrase the textbook's prototype case, where the minimum spanning tree problem occurs.



The graph above illustrates the road infrastructure that connects the stations in the Seervada Park. The distance in miles is shown by the number on each route. The park management has a proposal to run telephone lines beneath the roadways to connect the stations together. Just enough lines will be placed for connection between every pair of stations due to cost and environmental considerations, and the management is concerned about the minimum length of lines required to complete this work.

4 The Problem

Let's consider a network G = (N, A) with **n** nodes and **m** arcs called potential links and the arcs are undirected. Let X_{ij} be an arc belonging to A, associated with a length $X_{ij} > 0$. Our goal is to find a spanning tree T=(N, A'), a connected acycle subnetwork linking all the nodes in G with the minimum total arc length:

$$\min \mathbf{L}(\mathbf{T}) = \sum_{(i,j)\in \mathbf{A}'} x_{ij}$$

5 Prim's Algorithm

We now provide a method for finding an MST from an undirected and linked network. $\mathcal{G} = (\mathcal{N}, \mathcal{A})$. This algorithm is due to Prim [2].

- Step 1: Pick any node i in the network and set up an initial tree $\mathcal{T}_0 = (\mathcal{N}', \mathcal{A}') = (\{i\}, \emptyset)$.
- Step 2: If $\mathcal{N}' = \mathcal{N}$, the algorithm stops with the optimal total length $\ell(\mathcal{T})$. Otherwise, go to step 3.
- Step 3: Choose any pair of nodes $j_0 \in \mathcal{N} \setminus \mathcal{N}'$ and $i_0 \in \mathcal{N}'$ that satisfy

  $$x_{i_0 j_0} = \min\{x_{ij} : (i,j) \in \mathcal{A}, i \in \mathcal{N}', j \in \mathcal{N} \setminus \mathcal{N}'\}.$$

  If there are two or more satisfied pairs of nodes, then choose any one of them.
- Step 4: Update the new tree

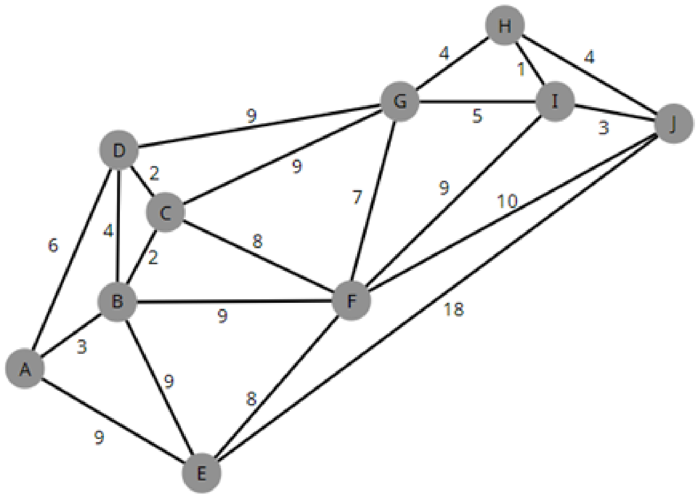
  $$\overline{\mathcal{T}} = (\overline{\mathcal{N}'} = \mathcal{N}' \cup \{j_0\}, \overline{\mathcal{A}'} = \mathcal{A}' \cup \{(i_0, j_0)\}).$$

  Return to step 2.

In other words, Prim's method starts at any node of the original network and keeps adding the nearest new (unselected) node reachable from the old (selected) nodes to the tree. The procedure terminates after precisely $n - 1$ iterations if the tree has n nodes.

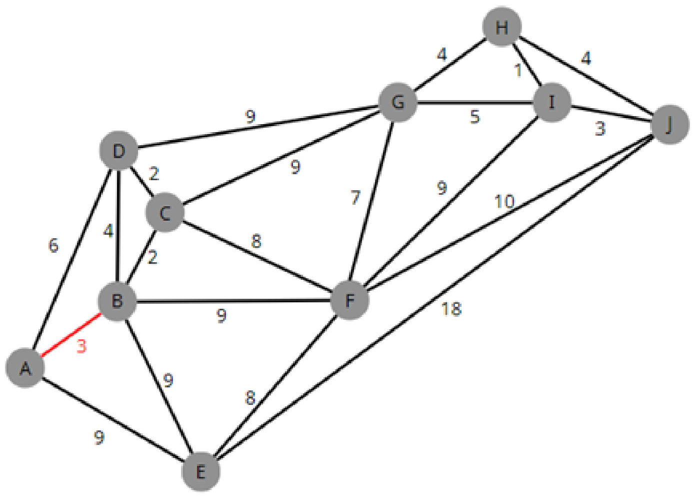
6 Graphical Visualization

We now apply Prim's algorithm on the minimum spanning tree problem given by our lecturer.



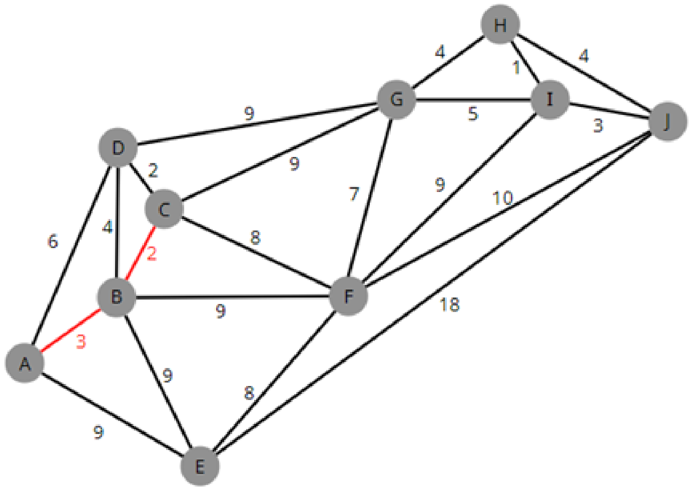
- Arbitrarily select node A to start. Our initial tree $\mathcal{T}_0 = (\{A\}, \emptyset)$ has one node, A, and no arc.

- The unconnected nodes accessible from A are B, D and E. Among the arcs AB, AD, AE, the one with the shortest length is AB with length 2, so we add the arc AB and the node B to $\mathcal{T}_0$ and get the updated tree $\mathcal{T}_1$.



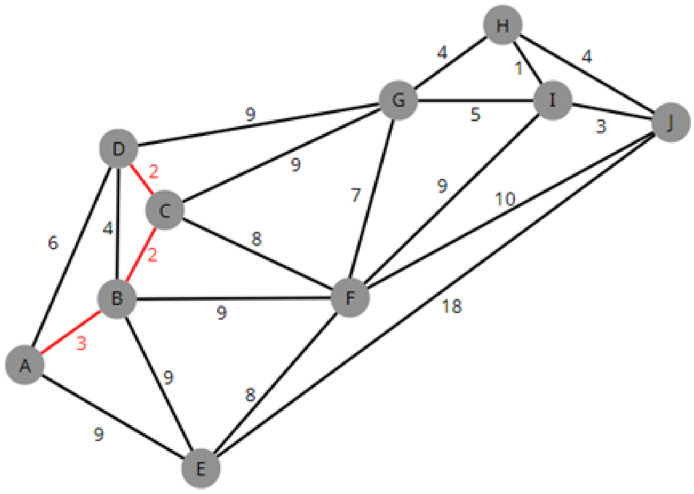
$\mathcal{T}_1 = (\{A, B\}, \{(A, B)\})$ with nodes A, B and arc AB.

- Now the unconnected nodes accessible from nodes in $\mathcal{T}_1$ are D, C, E, F. The shortest arc among BD, BC, BE, BF is BC (of length 2), so we add it and node C and the arc BC to $\mathcal{T}_1$ and get $\mathcal{T}_2$.



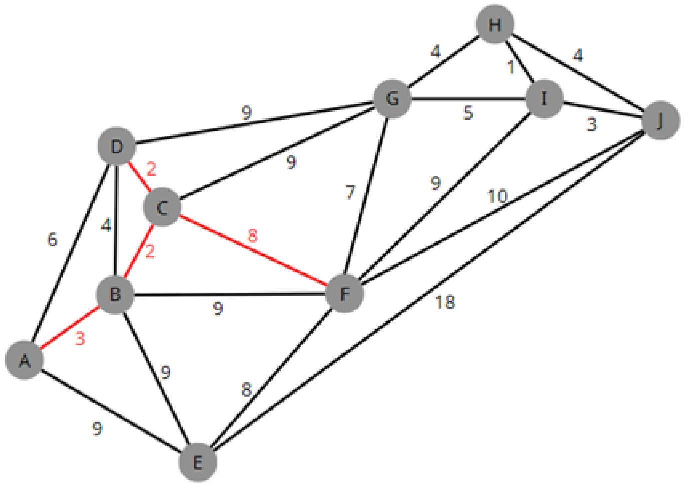
$$\mathcal{T}_2 = (\{A, B, C\}, \{(A, B), (B, C)\}).$$

- The unconnected, accessible nodes are D, G, F. The minimal arc is now CD, of length 2.



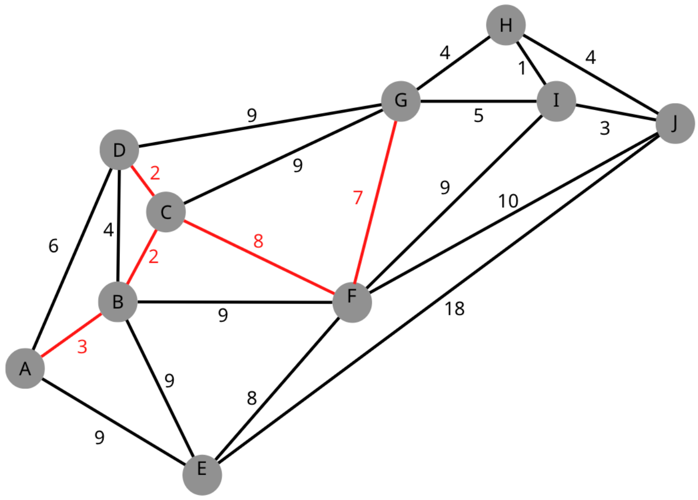
$$\mathcal{T}_3 = (\{A, B, C, D\}, \{(A, B), (B, C), (C, D)\}).$$

- The unconnected, accessible nodes are G, F. The minimal arc is now CF, of length 8.



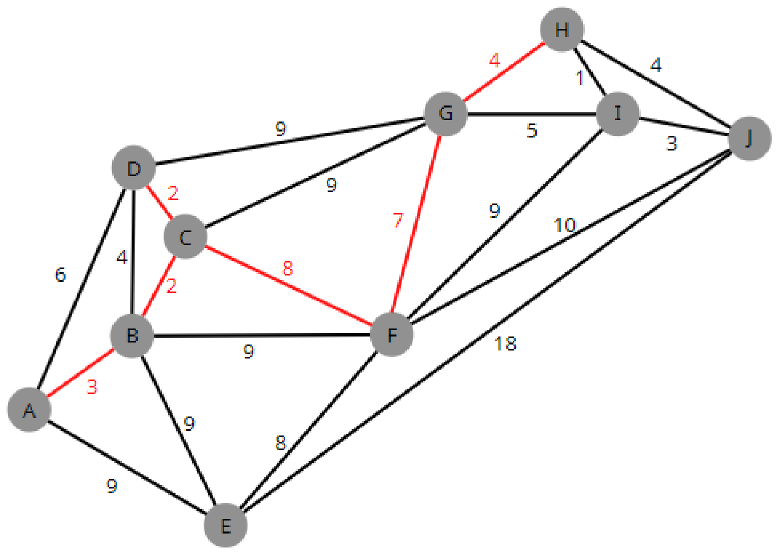
$$\mathcal{T}_4 = (\{A, B, C, D, F\}, \{(A, B), (B, C), (C, D), (C, F)\}).$$

- The unconnected, accessible nodes are E, G, I, J. The minimal arc is now FG, of length 7.



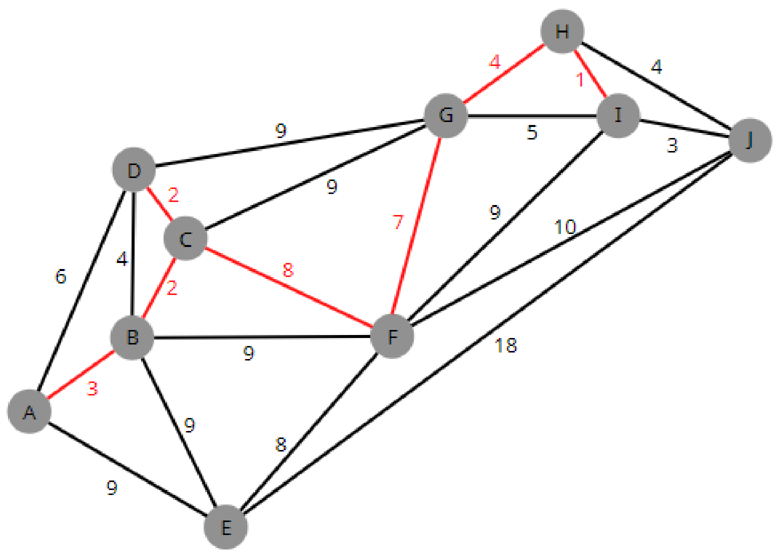
$$\mathcal{T}_5 = (\{A, B, C, D, F, G\}, \{(A, B), (B, C), (C, D), (C, F), (F, G)\}).$$

- The unconnected, accessible nodes are H, I. The minimal arc is now GH, of length 4.



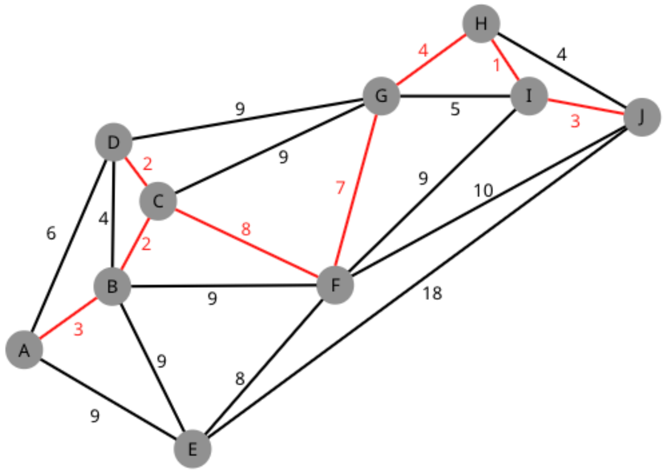
$\mathcal{T}_6(\{A, B, C, D, F, G, H\}, \{(A, B), (B, C), (C, D), (C, F), (F, G), (G, H)\})$.

- The unconnected, accessible nodes are I, J. The minimal arc is now HI, of length 1.



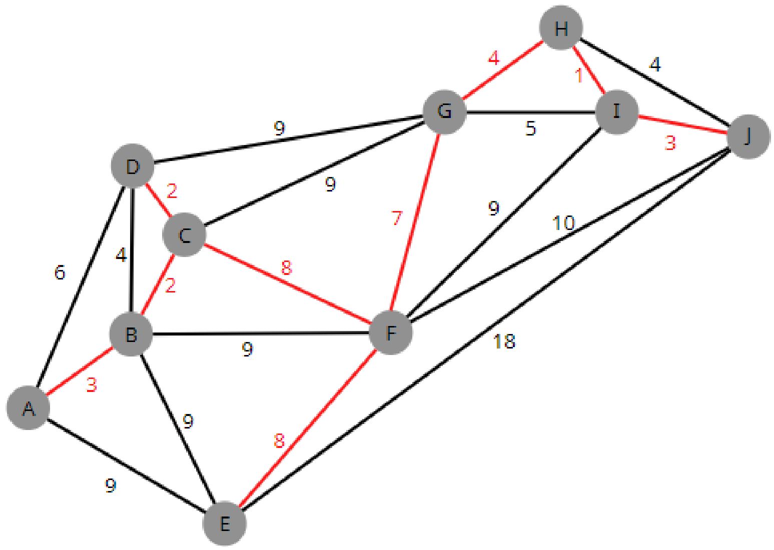
$\mathcal{T}_7(\{A, B, C, D, F, G, H, I\}, \{(A, B), (B, C), (C, D), (C, F), (F, G), (G, H), (H, I)\})$.

- The unconnected, accessible nodes are J, E. The minimal arc is now IJ, of length 3.



$\mathcal{T}_8 = (\{A, B, C, D, F, G, H, I, E\}, \{(A, B), (B, C), (C, D), (C, F), (F, G), (G, H), (H, I), (I, J)\})$.

- The only remaining unconnected node is E. The minimal arc is now FE, of length 8.



$\mathcal{T}_9 = (\{A, B, C, D, F, G, E, H, I, J\}, \{(A, B), (B, C), (C, D), (C, F), (F, G), (G, H), (H, I), (I, J), (F, E)\})$.

All nodes are now connected, so the algorithm stops with the optimal solution tree above and the optimal value of 38 (miles).

7 Remarks

We conclude this report by stating several remarks concerning on the MST problem.

- Multiple optimum solution trees may exist for some MST issues, which is typically the case when the beginning graph has arcs of the same length. However, the overall arc length of these trees ought to be the same.

- The MST problem is crucial to solving many real-world issues, particularly when deciding whether networks should be classified as electrical, communications, transportation, or oil/gas pipeline networks.

References

[1] F. S. Hillier, G. J. Lieberman, *Introduction to Operations Research, 7th ed.* McGraw-Hill, 2001.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein, *Introduction to Algorithms, 2nd ed.* MIT Press, 1990.