PRIM'S ALGORITHM

Team

SOMSHUBHRA GHOSH (21CE10066)

SREEJAN SHIVAM (21CE10067)

WORKFLOW

1.DEFINITION

- The definition of prims algorith
- Explanation of important terms

2.HOW IT IS IMPLEMENTED

- An Illustration is taken which has sequential steps to follow to achieve the desired result.

3.PROBLEM STATEMENT AND IT'S SOLUTION

- Problem statement is taken and its approach to solve is explained.

DEFINITION

PRIM'S ALGORITHM is a greedy algorithm that finds minimum spanning tree for a weighted undirected graph. It is also known as Jamik's algorithms.

TERMS TO DISCUSS !

GREEDY ALGORITHM

Weighted Undirected Graph

MST
(Minimum Spanning Tree)

GREEDY ALGORITHM

A greedy algorithm is an approach for solving a problem by selecting the best option available at the moment. It dosen't worry whether the current best result will bring the overall optimal result.

Advantages of Greedy Approach :
1.The algorithm is easier to describe.
2.This algorithm can perform better than other algorithm (but, not in all cases)

WEIGHTED UNDIRECTED GRAPH

WHAT IS GRAPH?

A Graph is a non-linear data structure consisting of vertices and edges. The vertices are sometimes also referred to as nodes and the edges are lines or arcs that connect any two nodes in the graph. More formally a Graph is composed of a set of vertices(V) and a set of edges(E). The graph is denoted by G(E, V)

Graphs are used to solve many real-life problems. Graphs are used to represent networks. The networks may include paths in a city or telephone network or circuit network. Graphs are also used in social networks like linkedIn, Facebook. For example, in Facebook, each person is represented with a vertex(or node). Each node is a structure and contains information like person id, name, gender, locale etc.

If the edges between the nodes are undirected, the graph is called an undirected graph. A weighted graph is a graph in which a number (the weight) is assigned to each edge. A graph is acyclic if it has no loop

MST (Minimum Spanning Tree)

A minimum spanning tree (MST) or minimum weight spanning tree is a subset of the edges of a connected edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight.

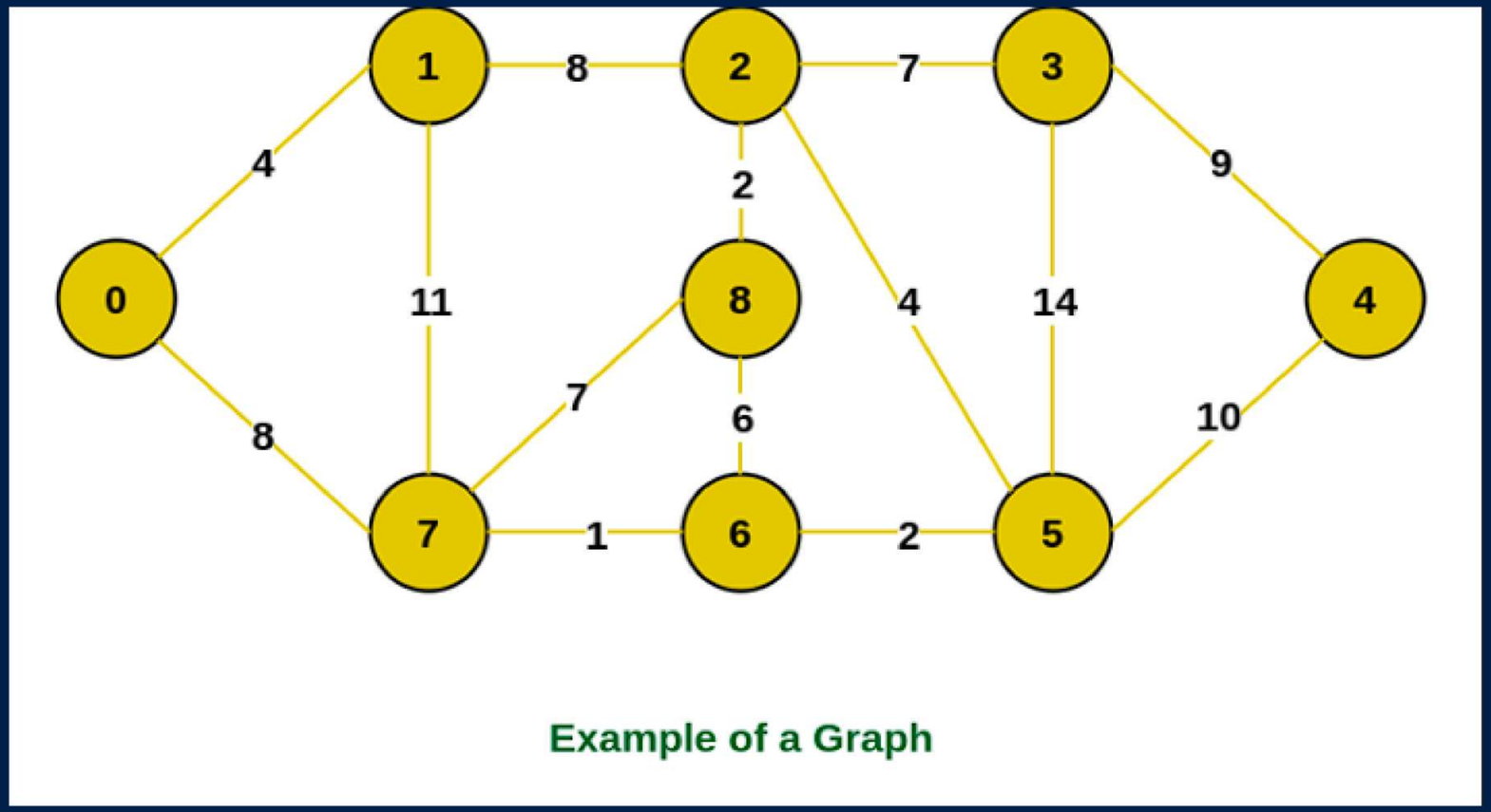
PROPERTIES OF SPANNING TREE:

- Spanning tree should not contain cycle, it should not be disconnected.
- Removing one edge from the spanning tree will make it disconnected.
- Adding one edge to the spanning tree will create a loop.
- If each edge has distinct weight then there will be only one MST.
- A complete connected graph can have n^(n-2) number of spanning tree.(n= no. of vertices
- Every connected and undirected graph has atleast one ST.
- From a complete graph by removing (e-n+1) edges we can get a ST (e= no. of edges, n= no. of vertices)
- A ST has same number of vertices as graph
- The number of edges in a ST is atmost (|V|-1).

HOW IT IS IMPLEMENTED

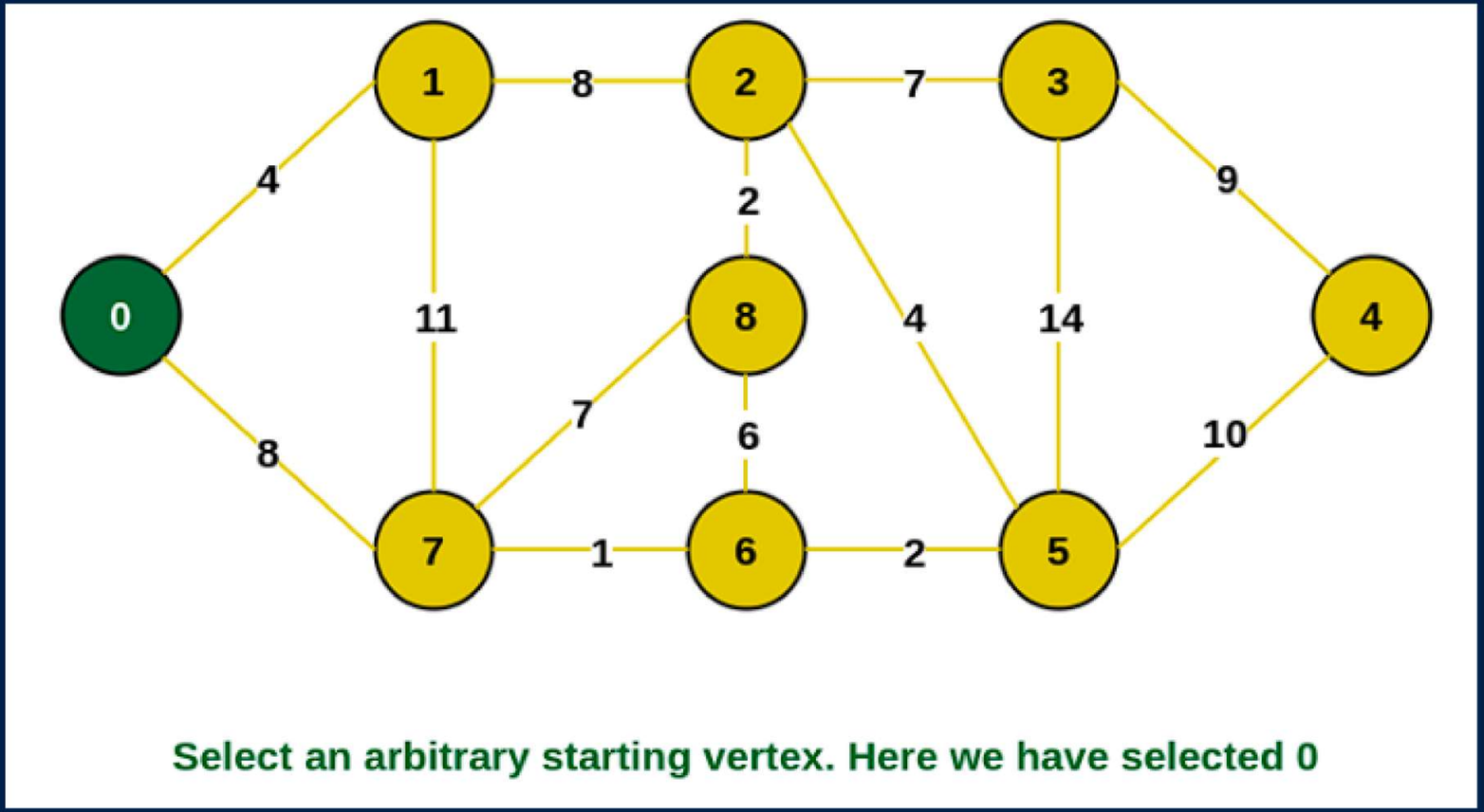
Let's understand this with the help of an illustration

Consider the following graph as an example for which we need to find the Minimum Spanning Tree (MST).

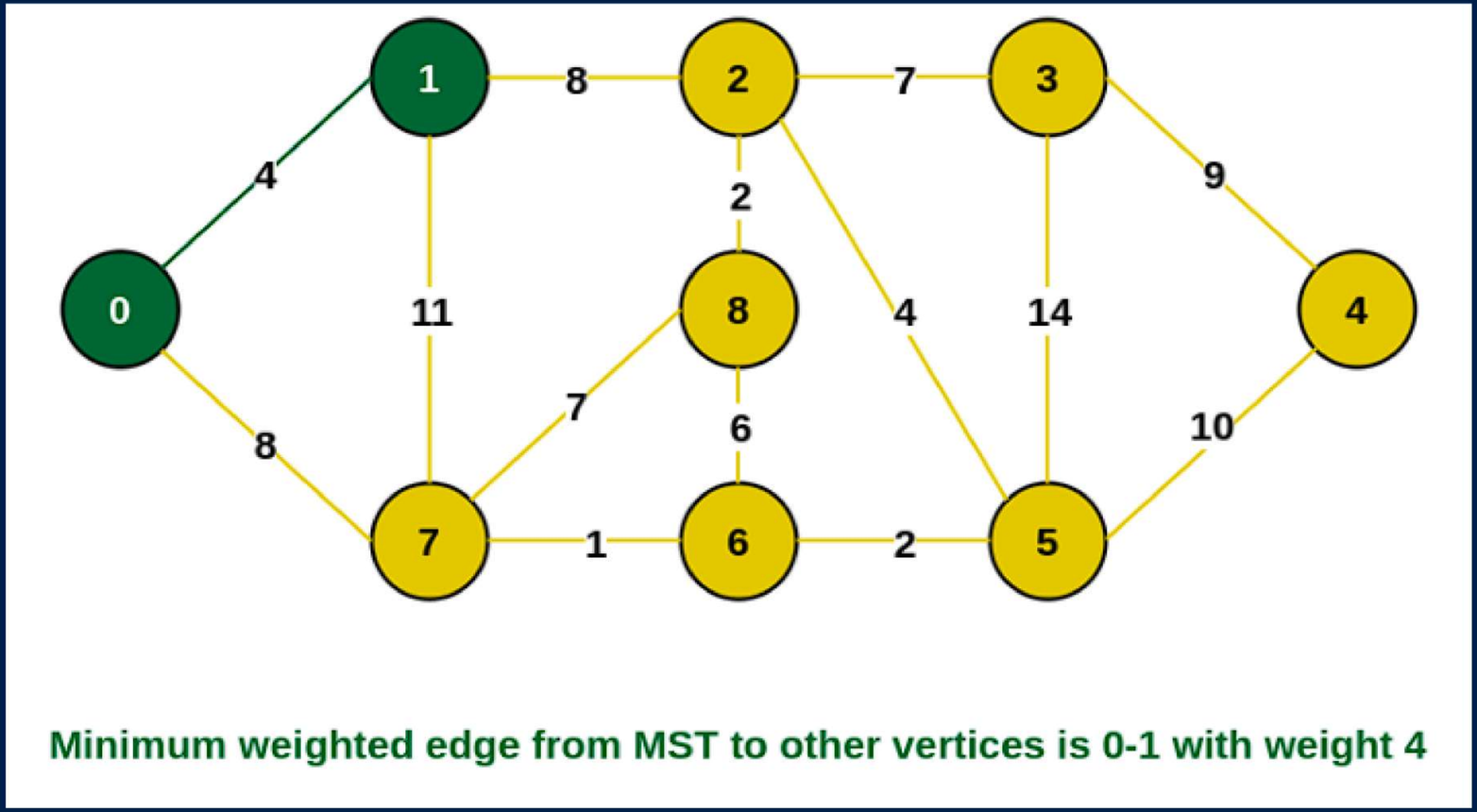


Example of a Graph

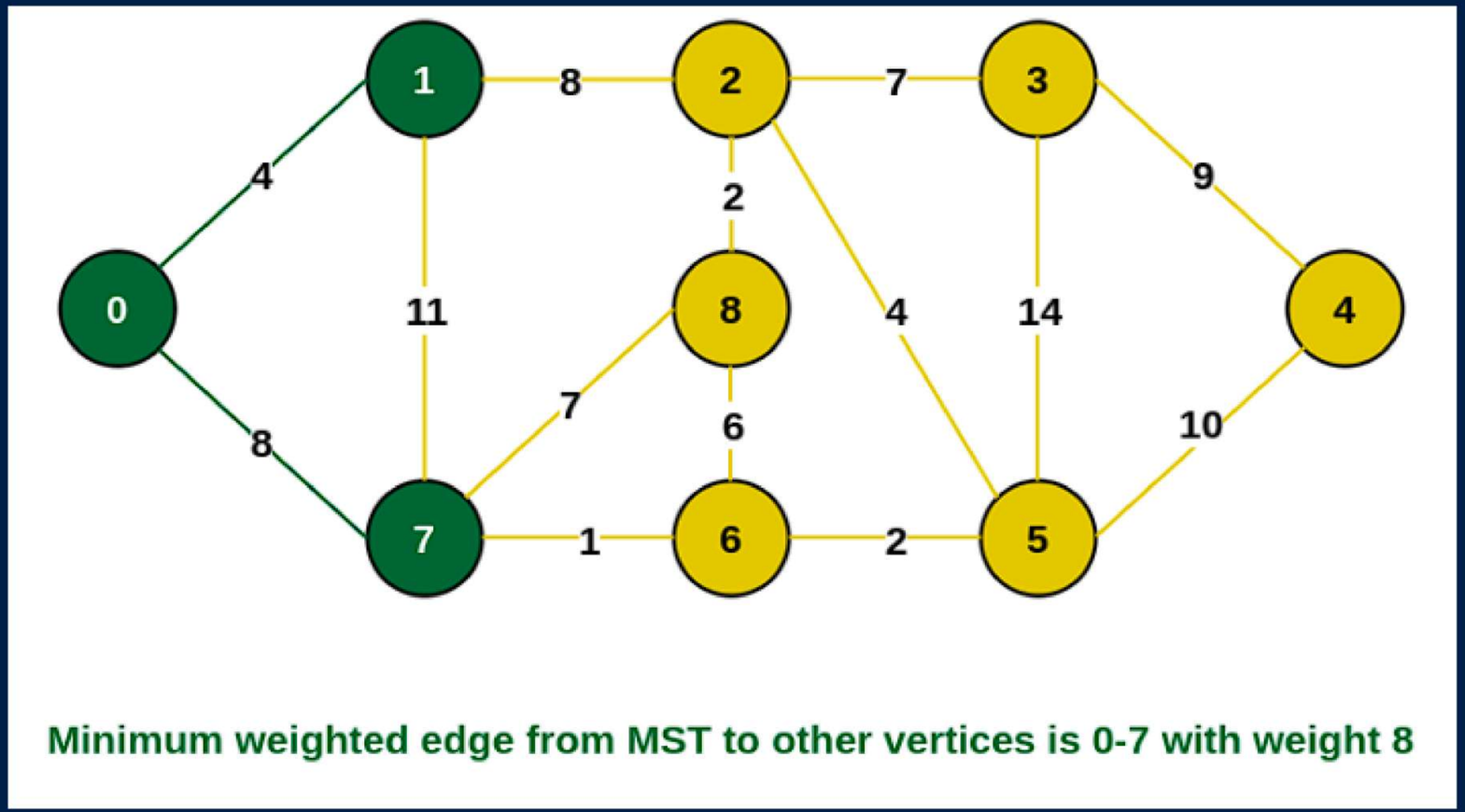
Step 1: Firstly, we select an arbitrary vertex that acts as the starting vertex of the Minimum Spanning Tree. Here we have selected vertex 0 as the starting vertex.



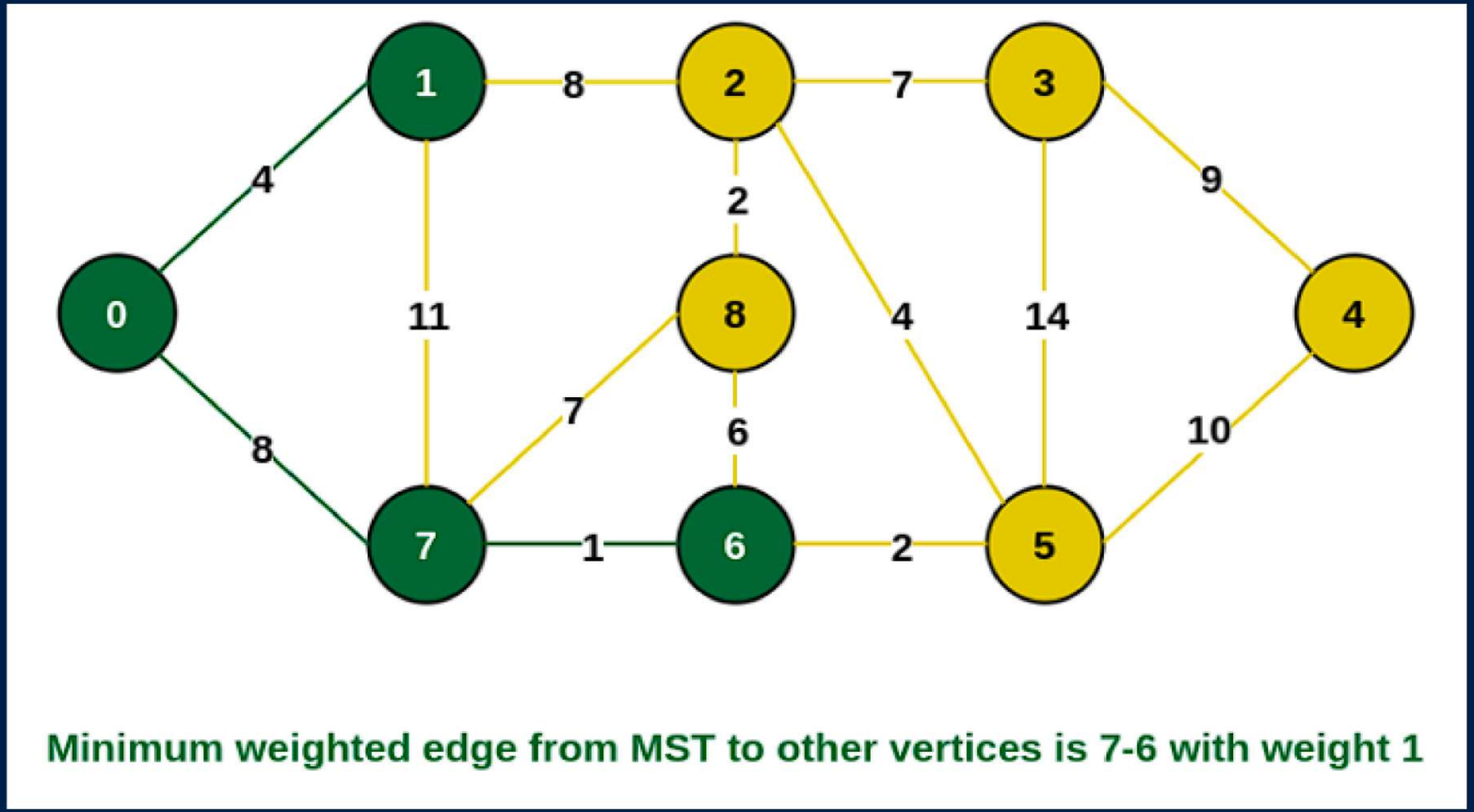
Step 2: All the edges connecting the incomplete MST and other vertices are the edges {0, 1} and {0, 7}. Between these two the edge with minimum weight is {0, 1}. So include the edge and vertex 1 in the MST.



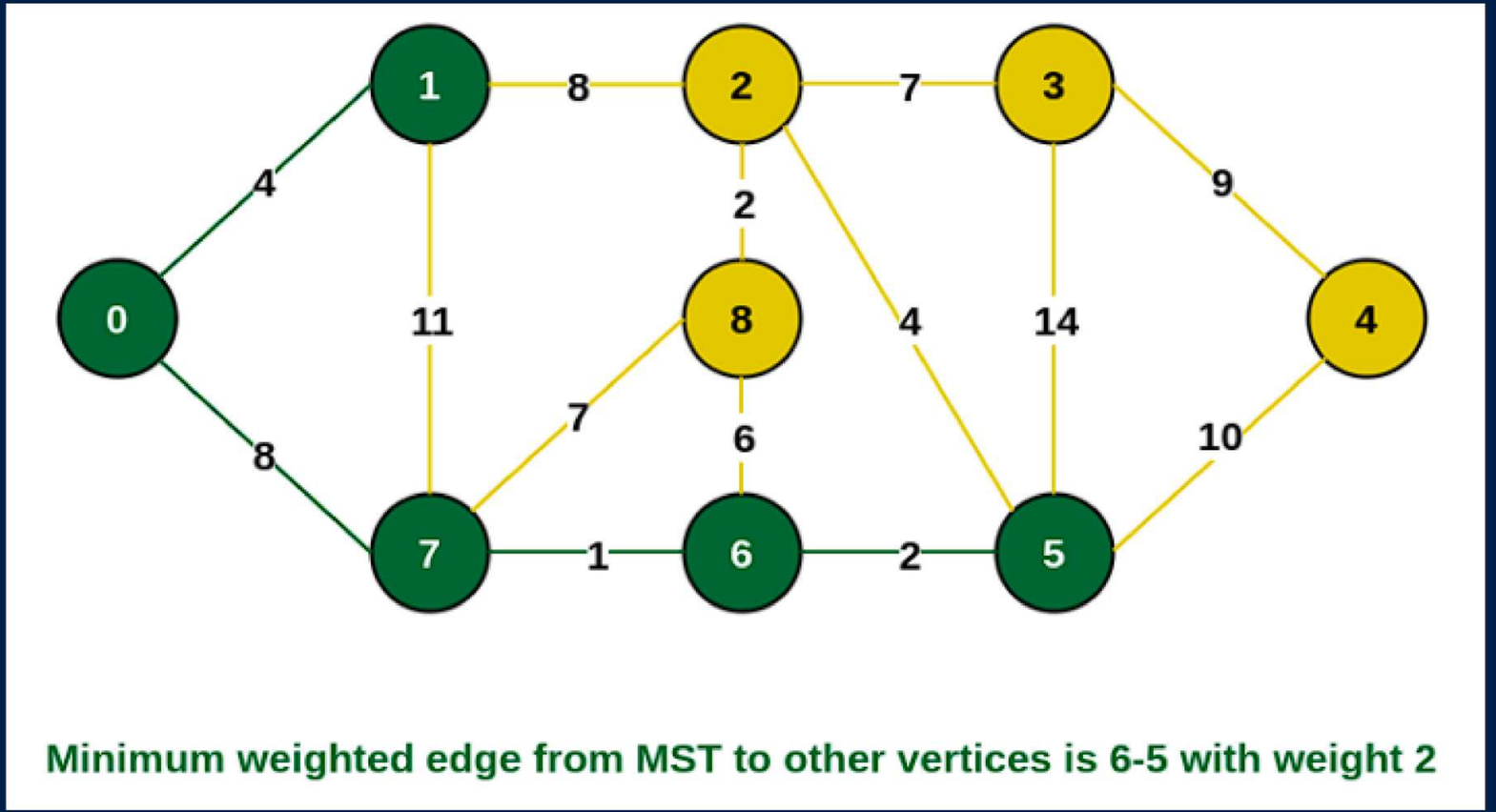
Step 3: The edges connecting the incomplete MST to other vertices are {0, 7}, {1, 7} and {1, 2}. Among these edges the minimum weight is 8 which is of the edges {0, 7} and {1, 2}. Let us here include the edge {0, 7} and the vertex 7 in the MST. [We could have also included edge {1, 2} and vertex 2 in the MST].



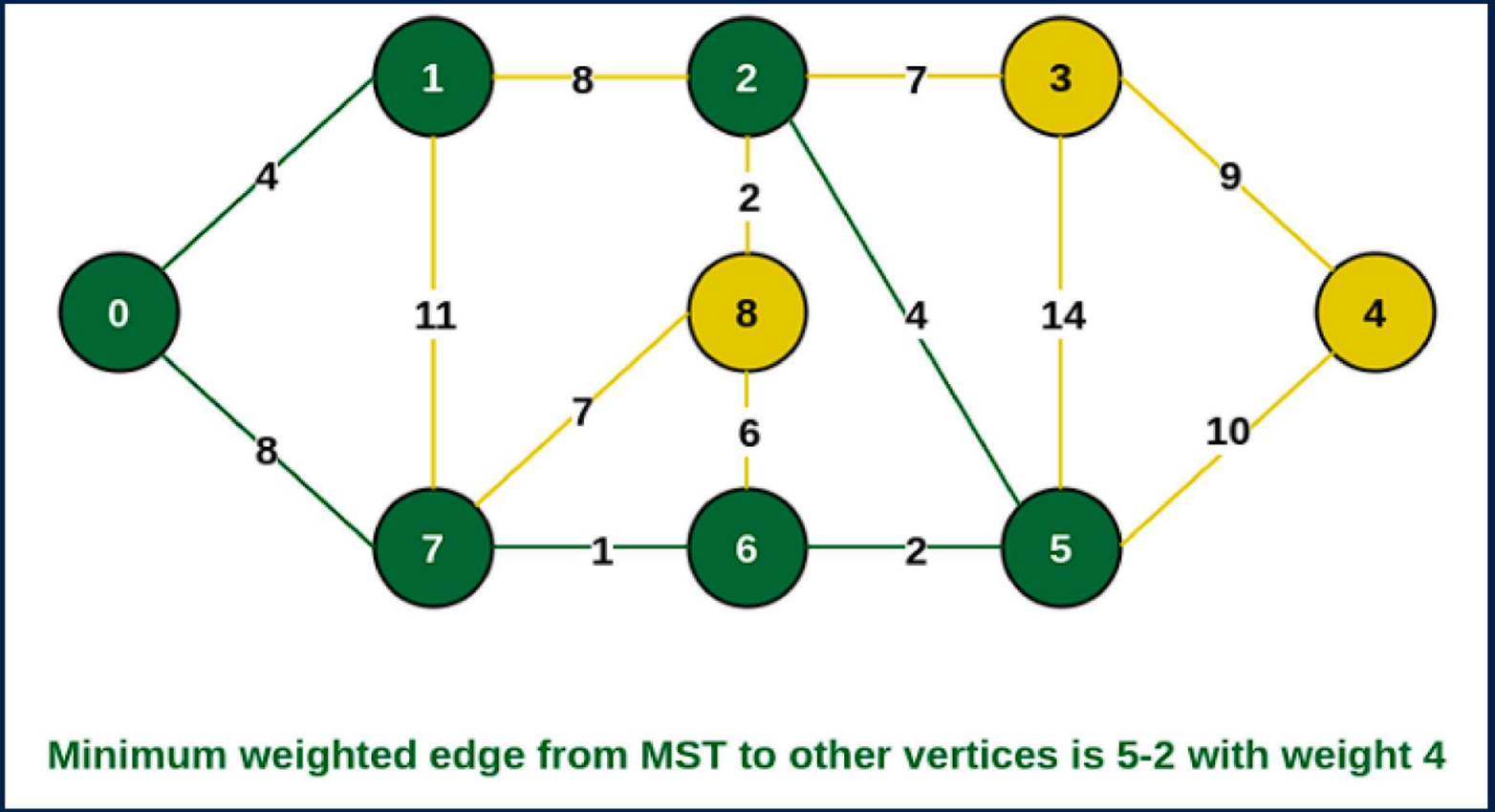
Step 4: The edges that connect the incomplete MST with the fringe vertices are {1, 2}, {7, 6} and {7, 8}. Add the edge {7, 6} and the vertex 6 in the MST as it has the least weight (i.e., 1).



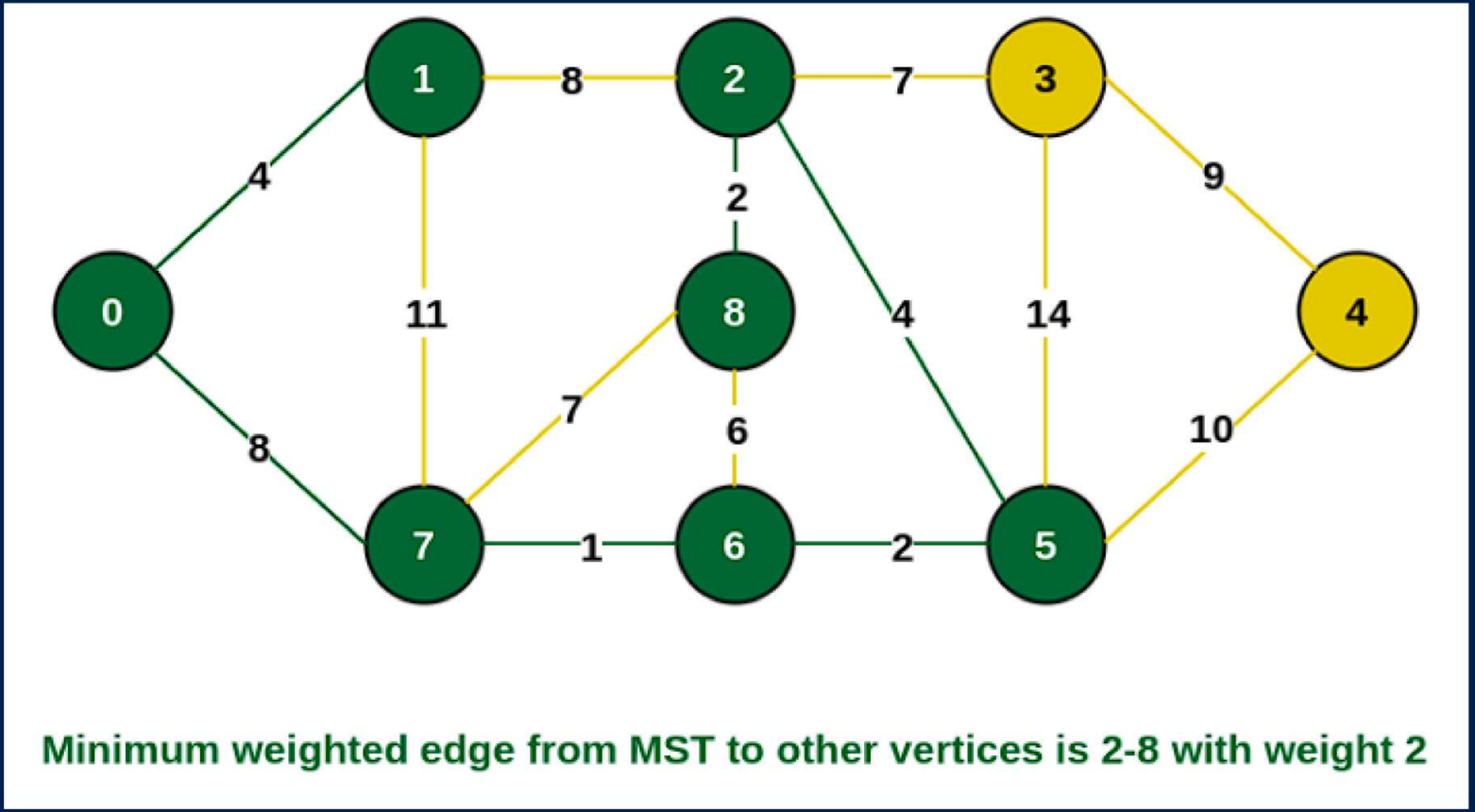
Step 5: The connecting edges now are {7, 8}, {1, 2}, {6, 8} and {6, 5}. Include edge {6, 5} and vertex 5 in the MST as the edge has the minimum weight (i.e., 2) among them.



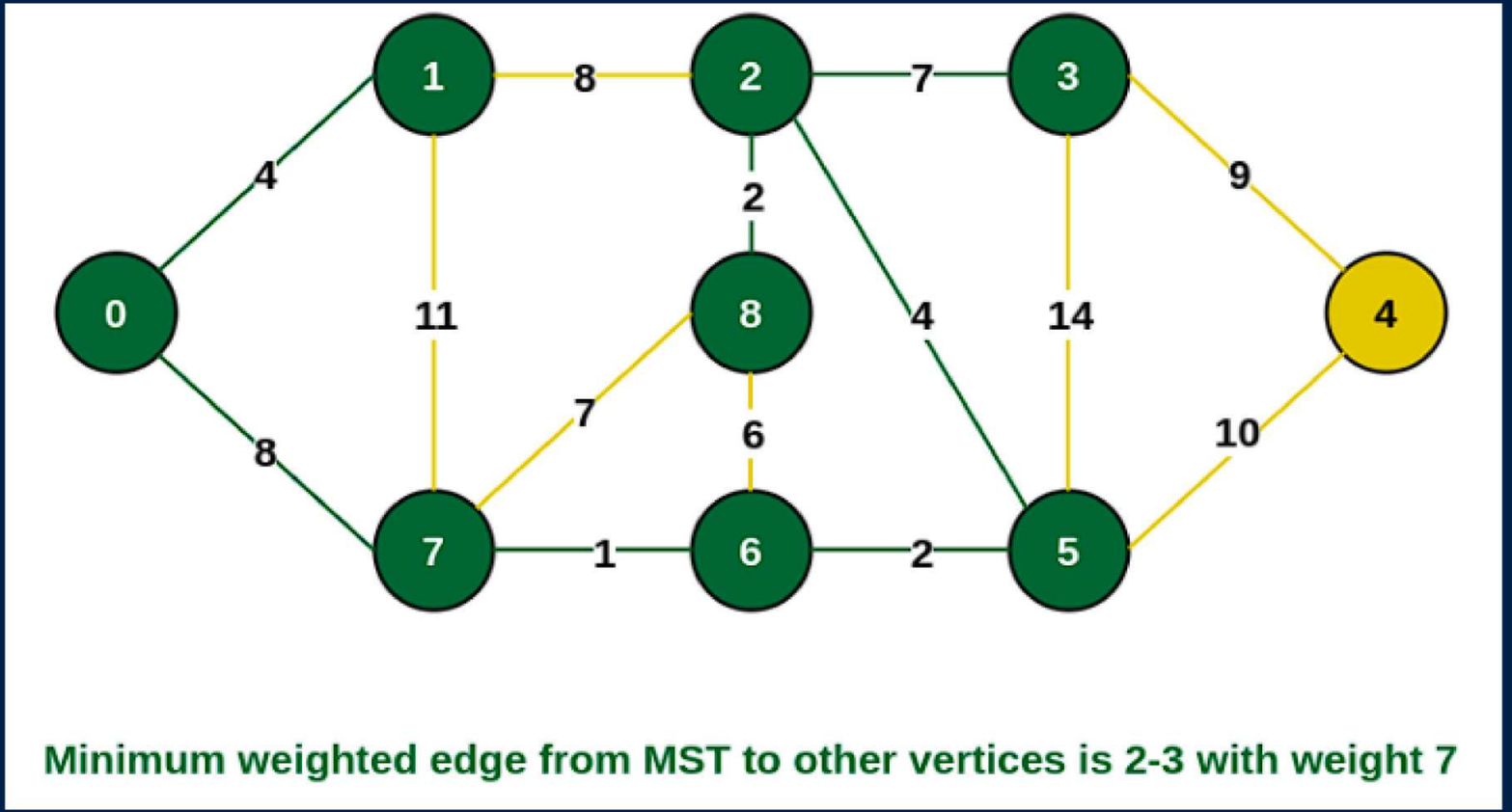
Step 6: Among the current connecting edges, the edge {5, 2} has the minimum weight. So include that edge and the vertex 2 in the MST.



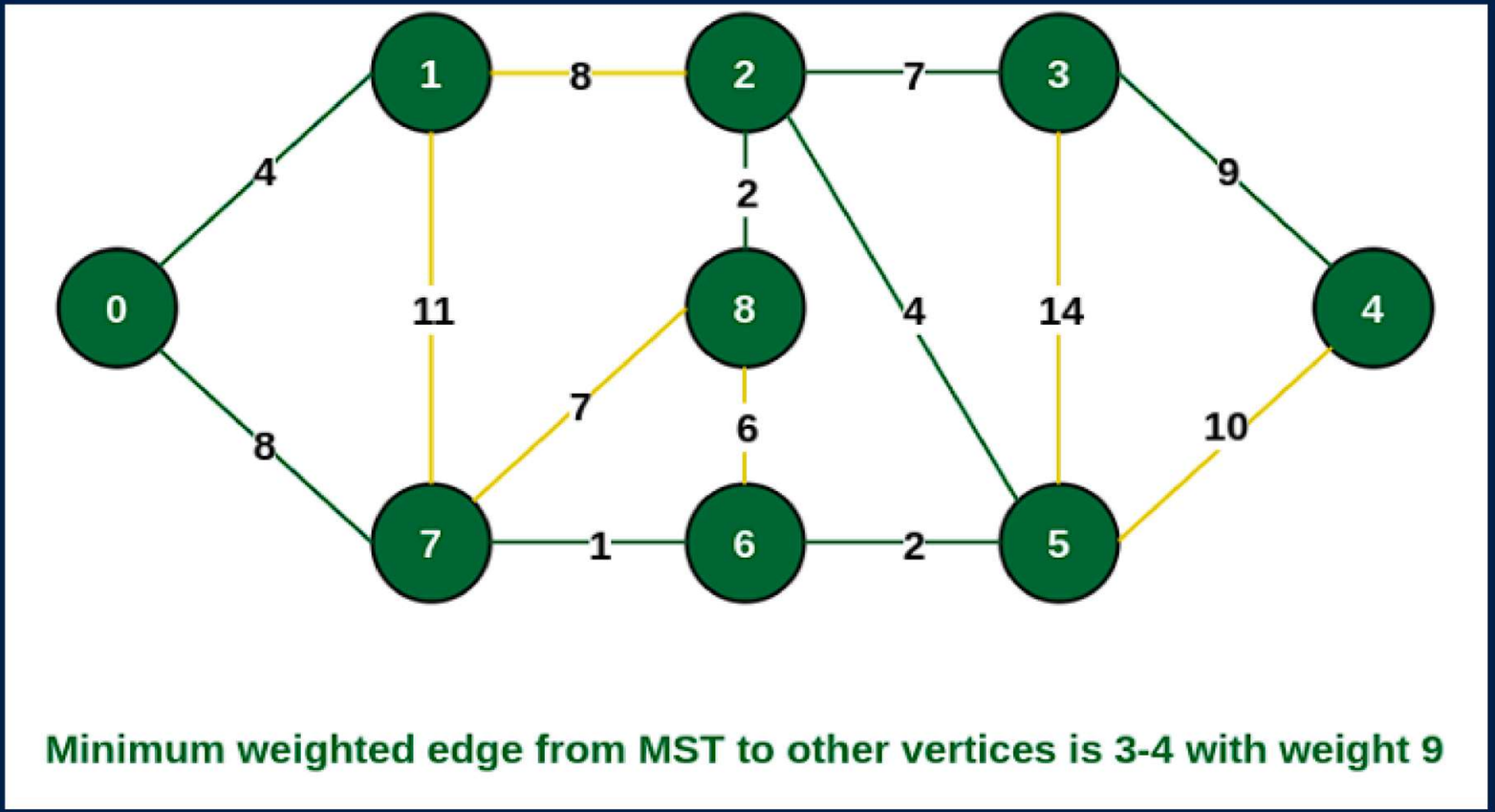
Step 7: The connecting edges between the incomplete MST and the other edges are {2, 8}, {2, 3}, {5, 3} and {5, 4}. The edge with minimum weight is edge {2, 8} which has weight 2. So include this edge and the vertex 8 in the MST.



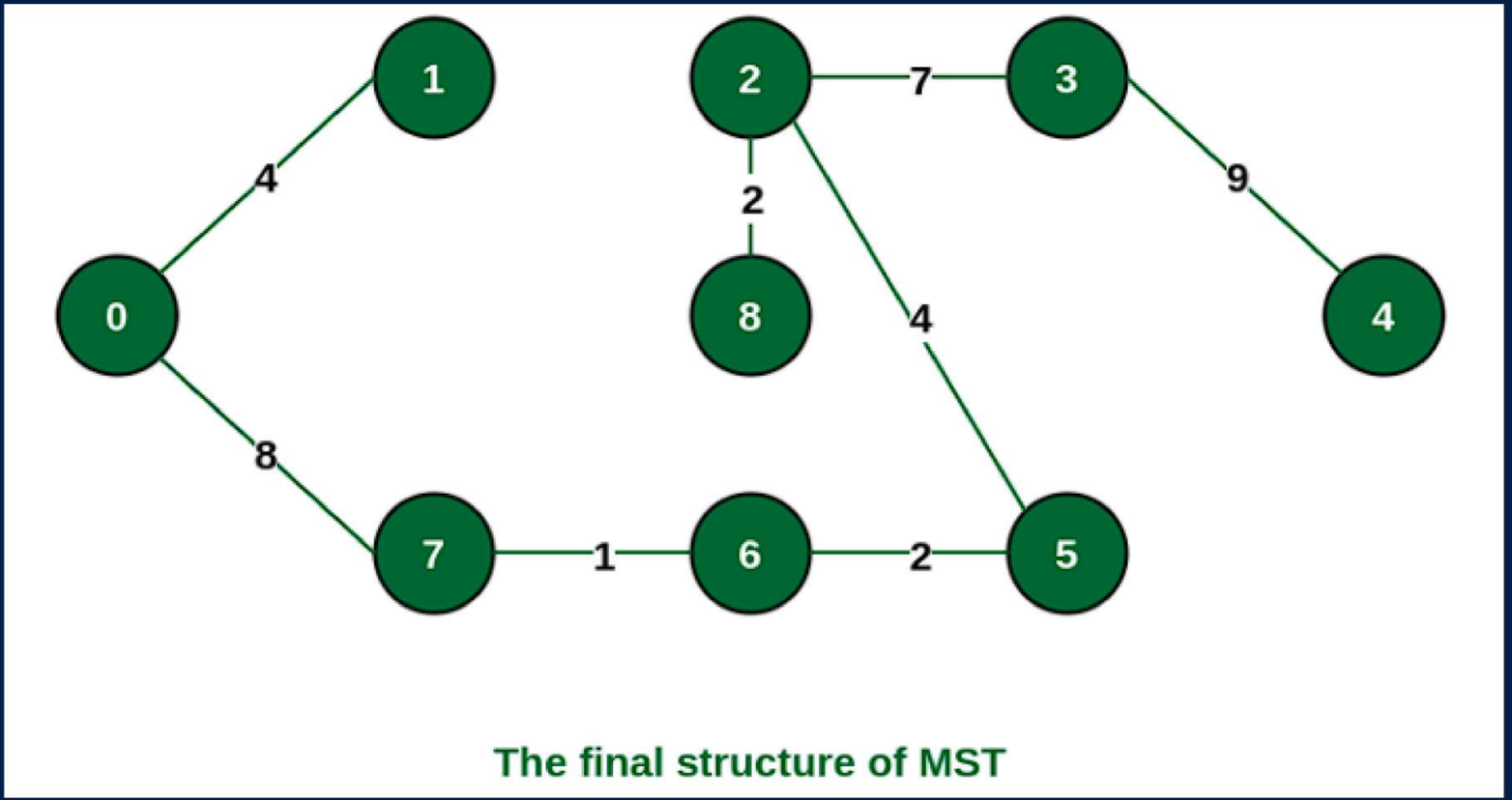
Step 8: See here that the edges {7, 8} and {2, 3} both have same weight which are minimum. But 7 is already part of MST. So we will consider the edge {2, 3} and include that edge and vertex 3 in the MST.



Step 9: Only the vertex 4 remains to be included. The minimum weighted edge from the incomplete MST to 4 is {3, 4}.

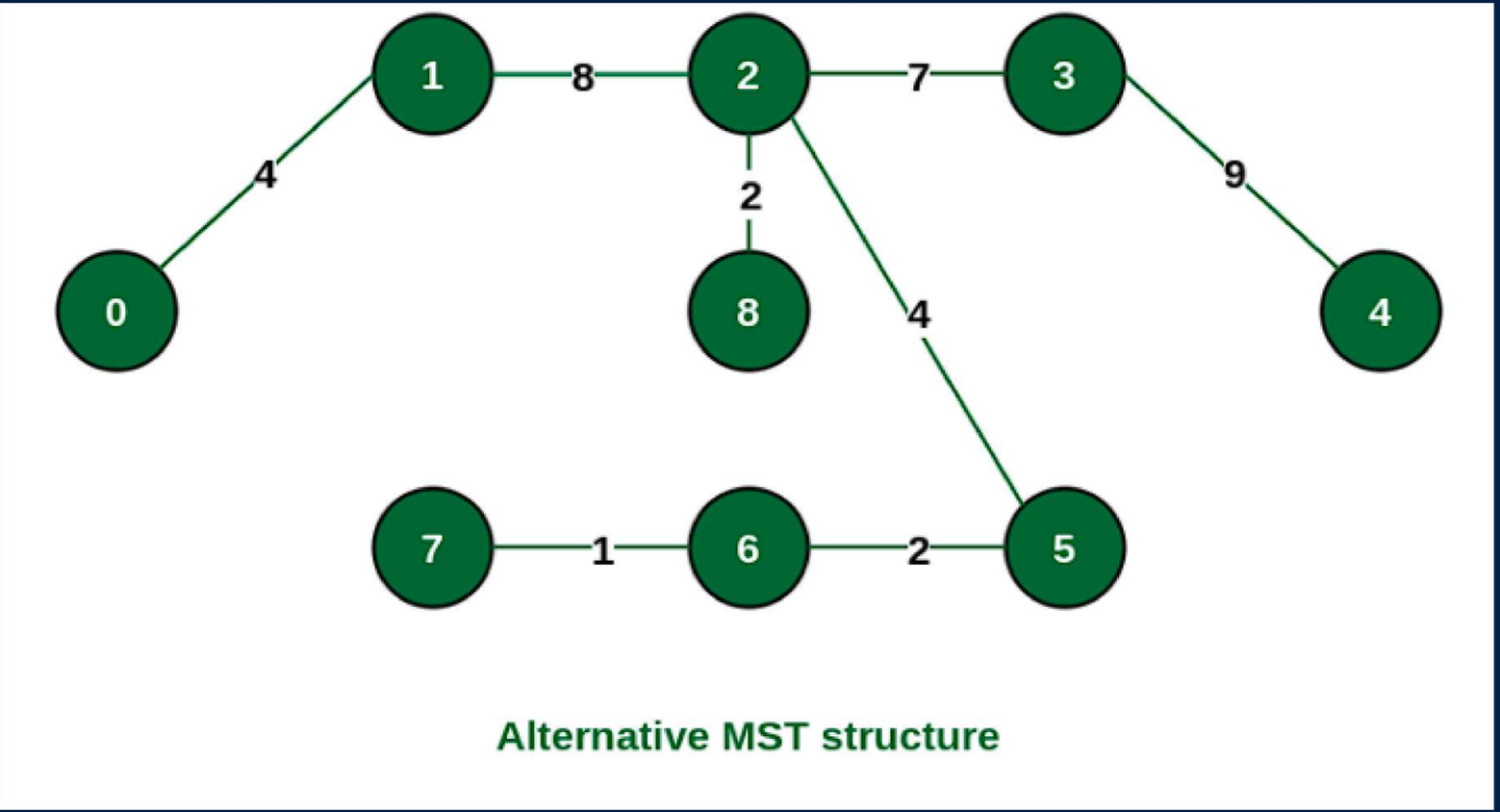


The final structure of the MST is as follows and the weight of the edges of the MST is (4 + 8 + 1 + 2 + 4 + 2 + 7 + 9) = 37.



The final structure of MST

Note: If we had selected the edge {1, 2} in the third step then the MST would look like the following.



Alternative MST structure

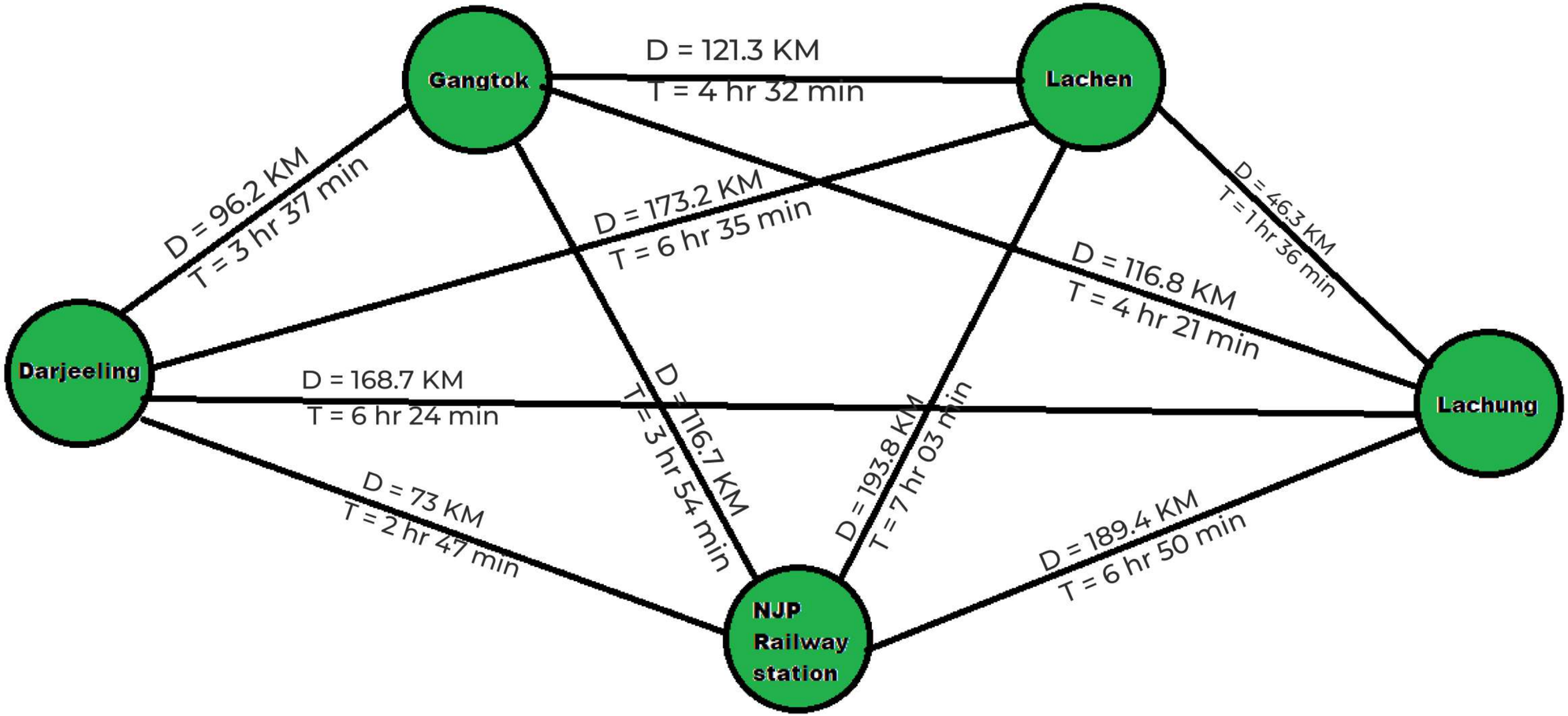
PROBLEM STATEMENT

A person wants to rent a car to travel to five places at a certain time of the day. He has the rates of two rental services. One rental service charges Rs. 210/- per hour while the second one charges Rs. 21/- per km. Formulate a code to find out which rental service should that person hire. The duration of travel between any two stations is assumed to be constant at that certain period of the day.

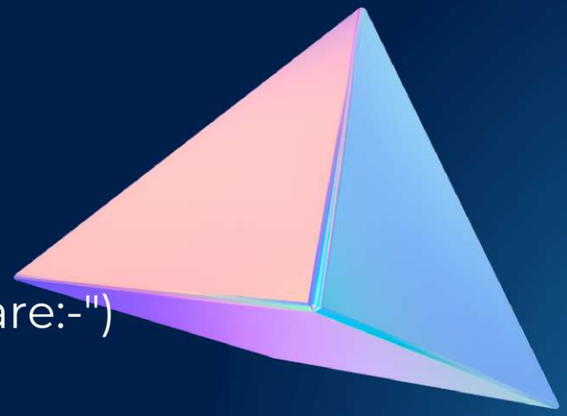
APPROACH:

- The program will be written in 2 parts.
- Part 1 will calculate the minimum time required to travel the places and will calculate the corresponding cost.
- Part 2 will find the minimum distance of travel and will give the cost.
- Both the costs will be compared and the rental service which charges lowest will be chosen

GRAPHICAL REPRESENTATION OF PROBLEM



CODE



```
print("Rental Services:-")
print("1. Cost = Rs. 210 per hour")
print("2. Cost = Rs. 21 per km")
print("Your distance and time graphs are:-")
V=5
time_gr=[[0,3,7,6,3],
    [3,0,5,4,3],
    [7,5,0,1,8],
    [6,4,1,0,7],
    [3,3,8,7,0]]
dist_gr=[[0,96,173,168,73],
    [96,0,121,117,116],
    [173,121,0,46,193],
    [168,117,46,0,189],
    [73,116,193,189,0]]
print("Time graph:-")
print(time_gr)
print("Distance graph:-")
print(dist_gr)
INF = 9999999
```

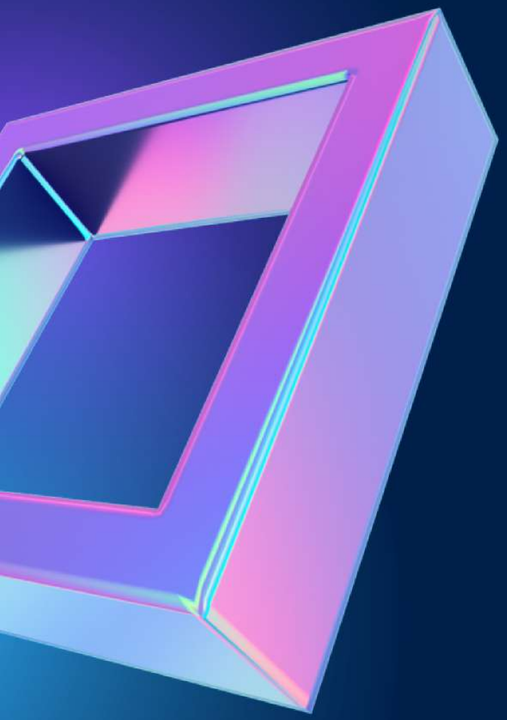
```
def time(time_gr):
  total_time=0
  selected = [0, 0, 0, 0, 0]
  no_edge = 0
  selected[0] = True
  while (no_edge < V - 1):
    minimum = INF
    x = 0
    y = 0
    for i in range(V):
      if selected[i]:
        for j in range(V):
          if ((not selected[j]) and time_gr[i][j]):
            if minimum > time_gr[i][j]:
              minimum = time_gr[i][j]
              x = i
              y = j
    total_time+=time_gr[x][y]
    selected[y] = True
    no_edge += 1
  return total_time
```

```
def distance(dist_gr):
  total_dist=0
  selected = [0, 0, 0, 0, 0]
  no_edge = 0
  selected[0] = True
  while (no_edge < V - 1):
    minimum = INF
    x = 0
    y = 0
    for i in range(V):
      if selected[i]:
        for j in range(V):
          if ((not selected[j]) and dist_gr[i][j]):
            if minimum > dist_gr[i][j]:
              minimum = dist_gr[i][j]
              x = i
              y = j
    total_dist+=dist_gr[x][y]
    selected[y] = True
    no_edge += 1
  return total_dist
```

```
total_time=time(time_gr)
total_dist=distance(dist_gr)
time_cost= 210*total_time
dist_cost= 21*total_dist
if(time_cost>dist_cost):
  print("Choose Rental Service 2")
else:
  print("Choose Rental Service 1")
```

OUTPUT

Rental Services:-
1. Cost = Rs. 210 per hour
2. Cost = Rs. 21 per km
Your distance and time graphs are:-
Time graph:-
[[0, 3, 7, 6, 3], [3, 0, 5, 4, 3], [7, 5, 0, 1, 8], [6, 4, 1, 0, 7], [3, 3, 8, 7, 0]]
Distance graph:-
[[0, 96, 173, 168, 73], [96, 0, 121, 117, 116], [173, 121, 0, 46, 193], [168, 117, 46, 0, 189], [73, 116, 193, 189, 0]]
Choose Rental Service 1



Thank You