# Prim’s algorithm

**Definition:**

1. Prim’s algorithm for finding the Minimum Spanning Tree (MST) for given undirected graph G (V, E) with weights.

**Prim’s Algorithm:**

1. Firstly, we take graph input as a matrix, for each edge between the vertices, we take weights as input from the below function. Once we have all the vertices and edge weights, we use prims algorithm to find the Minimum Spanning Tree (MST).

**Text

Description automatically generated with medium confidence**

1. We use Prim’s algorithm to find the Minimum Spanning Tree (MST), I have created an array “Output” to hold all the selected vertices. We initialize the edge count to zero. Also, the 0th vertex is made true. Now we loop in V-1 times, as the MST will have less than V vertices. For each vertex, we find the adjacent vertices and calculate the weights from the first node to the current node. If the vertex already exists in the “Output”, we discard that vertex and search for the nearest vertex to the selected vertex.

Text

Description automatically generated

**Pseudocode to implement Prim’s algorithm for finding the Minimum Spanning Tree (MST) for given undirected graph G (V, E) with weights.**

Text

Description automatically generated

Text

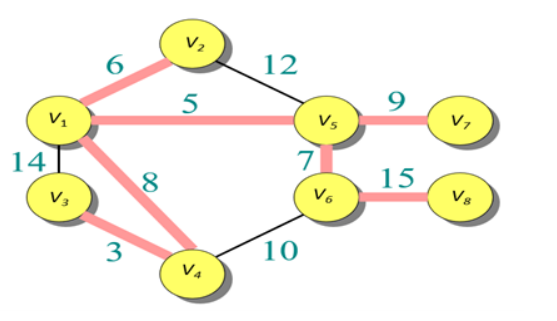
Description automatically generated with medium confidence

**Graphical user interface, application

Description automatically generated**

**Execution/ Output:**

1. Verify the correctness of your program by solving the problem of finding the MST for the following graph as explained in the class.



**Graphical user interface, application

Description automatically generated**

**A picture containing application

Description automatically generated**

**Text

Description automatically generated**

1. Verify the correctness of your program by solving the problem of finding the MST for the following graph as explained in the class.

Shape

Description automatically generated with low confidence

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, application, email

Description automatically generated**

1. Verify the correctness of your program by solving the problem of finding the MST for the following graph as explained in the class.

A picture containing text, watch

Description automatically generated

**Graphical user interface, application

Description automatically generated**

**Graphical user interface, text, email

Description automatically generated**