Implementation

```
# Prim's Algorithm in Python


INF = 9999999

# number of vertices in graph

V = 5

# create a 2d array of size 5x5

# for adjacency matrix to represent graph

G = [[0, 9, 75, 0, 0],

     [9, 0, 95, 19, 42],

     [75, 95, 0, 51, 66],

     [0, 19, 51, 0, 31],

     [0, 42, 66, 31, 0]]

# create a array to track selected vertex

# selected will become true otherwise false

selected = [0, 0, 0, 0, 0]

# set number of edge to 0

no_edge = 0

# the number of egde in minimum spanning tree will be

# always less than(V - 1), where V is number of vertices in

# graph

# choose 0th vertex and make it true

selected[0] = True

# print for edge and weight

print("Edge : Weight\n")

while (no_edge < V - 1):

    # For every vertex in the set S, find the all adjacent vertices
    #, calculate the distance from the vertex selected at step 1.
    # if the vertex is already in the set S, discard it otherwise
    # choose another vertex nearest to selected vertex  at step 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 G[i][j]):
                    # not in selected and there is an edge
                    if minimum > G[i][j]:
                        minimum = G[i][j]
                        x = i
                        y = j
    print(str(x) + "-" + str(y) + ":" + str(G[x][y]))
    selected[y] = True
    no_edge += 1
```

Output :

```
Edge : Weight

0-1:9
1-3:19
3-4:31
3-2:51
```