Objective: **Implementation and analysis of Prim's Algorithm**

Prim's Algorithm

Prim's algorithm is a minimum spanning tree algorithm that takes a graph as input and finds the subset of the edges of that graph which

- form a tree that includes every vertex
- has the minimum sum of weights among all the trees that can be formed from the graph

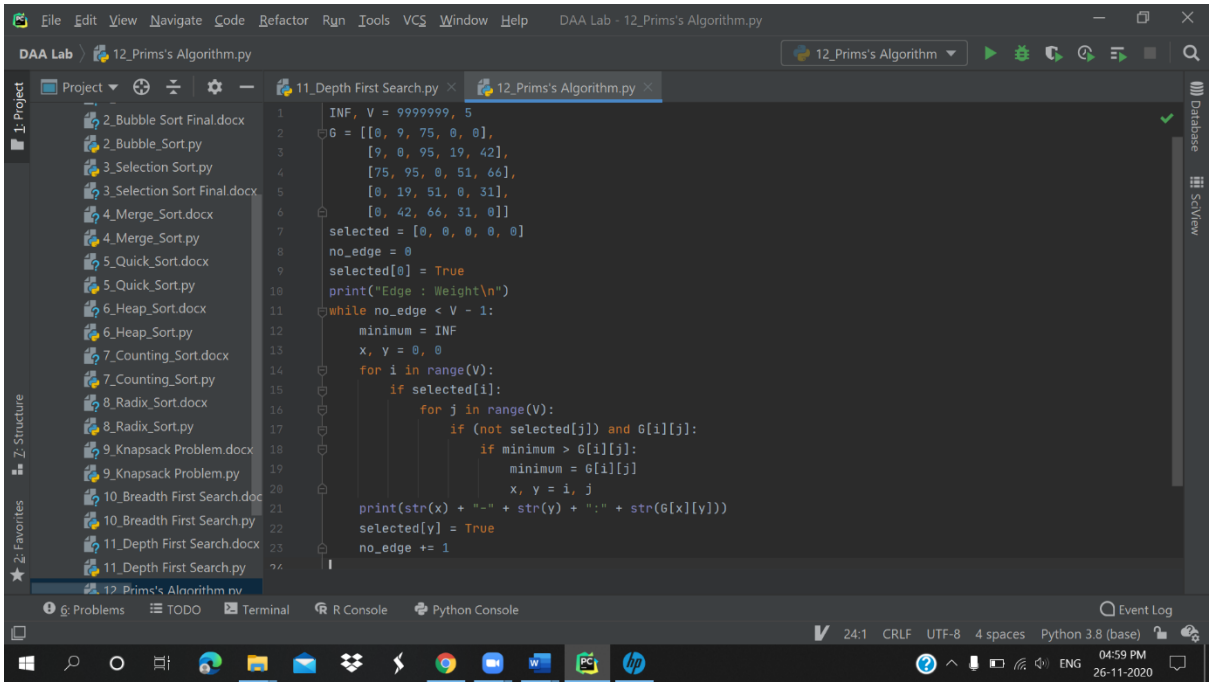
Prim's algorithm

The steps for implementing Prim's algorithm are as follows:

1. Initialize the minimum spanning tree with a vertex chosen at random.
2. Find all the edges that connect the tree to new vertices, find the minimum and add it to the tree
3. Keep repeating step 2 until we get a minimum spanning tree.

Code:

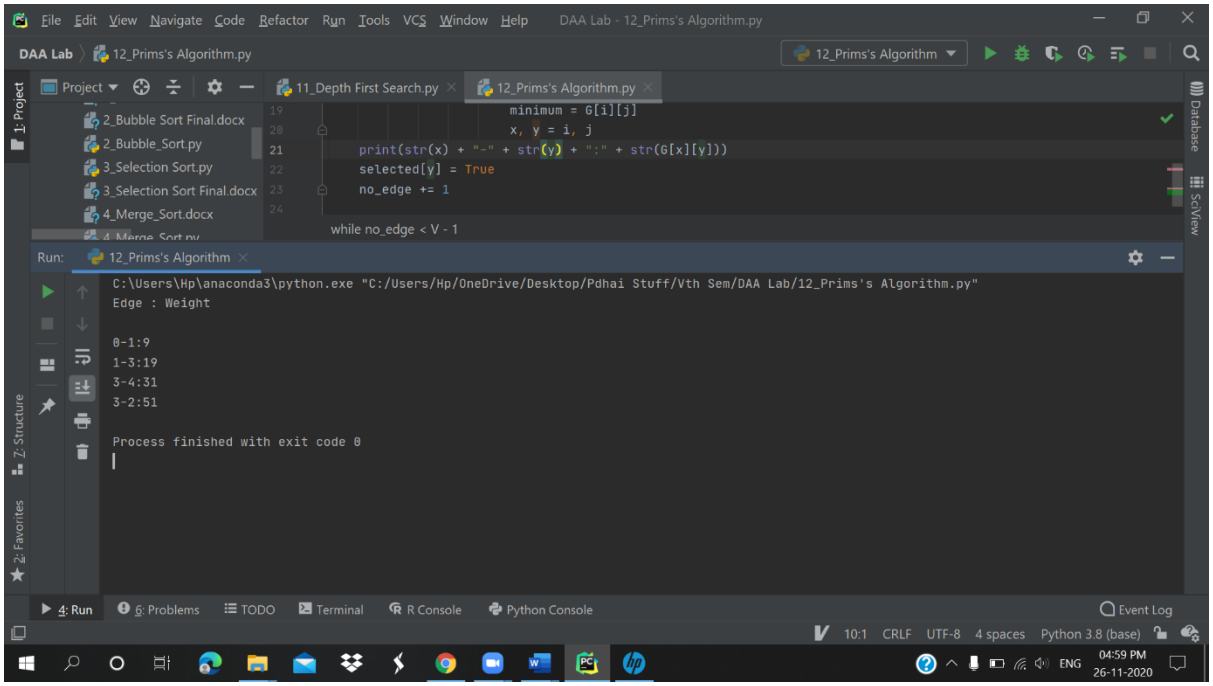
```
INF, V = 9999999, 5
G = [[0, 9, 75, 0, 0],
     [9, 0, 95, 19, 42],
     [75, 95, 0, 51, 66],
     [0, 19, 51, 0, 31],
     [0, 42, 66, 31, 0]]
selected = [0, 0, 0, 0, 0]
no_edge = 0
selected[0] = True
print("Edge : Weight\n")
while no_edge < V - 1:
    minimum = INF
    x, y = 0, 0
    for i in range(V):
        if selected[i]:
            for j in range(V):
                if (not selected[j]) and G[i][j]:
                    if minimum > G[i][j]:
                        minimum = G[i][j]
                        x, y = i, 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
```



Time Complexities:

The time complexity of Prim's algorithm is `O (E log V)`.

Prim's Algorithm Applications:

- Laying cables of electrical wiring
- In network designed
- To make protocols in network cycles