Date: Ex No: 2.3	Title of the Lab Travelling Salesman	Name: Yuvraj Singh Chauhan Registration Number: RA1911027010058 Section: N1 Lab Batch: 1 Day Order: 3

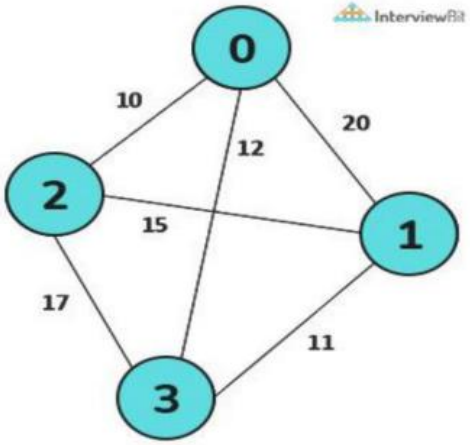
AIM:

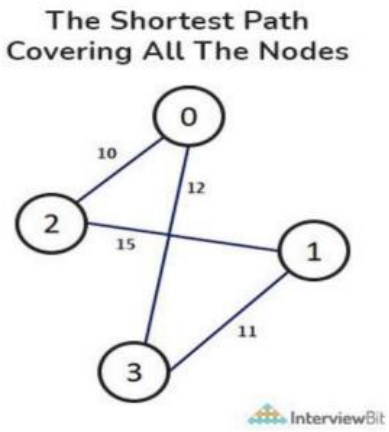
To implement Travelling Salesman Problem.

Description of the Concept or Problem given:

Given a set of cities and distance between every pair of cities, the problem is to find the shortest possible route that visits every city exactly once and returns to the starting point.

Manual Solution:





From node 0 the shortest distance is 10, so move to node 2.
From node 2 the shortest distance is 15, so move to node 1.
From node 1 the shortest distance is 11, so move to node 3
From node 3 the shortest distance is 12, so move back to node 0.
So, in this case the shortest possible route would be of 10+15+11+12=48

Program Implementation [Coding]

```
from sys import maxsize
from itertools import permutations
V = 4
def TSP(graph, s):
    vertex = []
    for i in range(V):
        if i != s:
            vertex.append(i)
    min_path = maxsize
    next_permutation=permutations(vertex)
    for i in next_permutation:
        current_pathweight = 0
        k = s
        for j in i:
```

```
            current_pathweight += graph[k][j]
            k = j
        current_pathweight += graph[k][s]
        min_path = min(min_path, current_pathweight)
    return min_path
graph = [[0,20,10,12],[20,0,15,11],[10,15,0,17],[12,11,17,0]]
d=0
print(TSP(graph,d))
```

Screenshots of the Outputs:

```
            k = j
        current_pathweight += graph[k][s]
        min_path = min(min_path, current_pathweight)
    return min_path
graph = [[0,20,10,12],[20,0,15,11],[10,15,0,17],[12,11,17,0]]
d=0
print(TSP(graph,d))
```

```
48
```

Signature of the Student

[YUVRAJ SINGH CHAUHAN]