



PORT CITY INTERNATIONAL UNIVERSITY

Course Code:

Course Title:

Report Name:

Submitted To:

Name of Lecturer :

Department :

Submitted By:

Name of Student : Mohammed Arsalan Morshed

Program : B.Sc. in CSE

Batch No : CSE-30-D-A2

ID No : CSE-03007824

Date of submission :

Experiment No: 06

Experiment Name: Finding minimum cost using Travelling Salesman Problem (TSP)

Code:

```
def calculate_cost(graph, tour, n):  
    cost = sum(graph[tour[i]][tour[i + 1]] for i in range(n - 1))  
    return cost + graph[tour[n - 1]][tour[0]]  
  
def tsp(graph, n, tour, visited, level, min_cost, best_tour):  
    if level == n:  
        current_cost = calculate_cost(graph, tour, n)  
        if current_cost < min_cost[0]:  
            min_cost[0] = current_cost  
            best_tour[:] = tour[:]  
        return  
  
    for i in range(n):  
        if not visited[i]:  
            visited[i], tour[level] = True, i  
            tsp(graph, n, tour, visited, level + 1, min_cost, best_tour)  
            visited[i], tour[level] = False, -1  
  
def main():  
    graph = [  
        [0, 30, 65, 20],  
        [30, 0, 20, 55],  
        [65, 20, 0, 40],  
        [20, 55, 40, 0]  
    ]  
    n = len(graph)
```

```
start_node = 0 # Change this to your desired starting node (0, 1, 2, etc.)

tour = [-1] * n
visited = [False] * n
min_cost = [float('inf')]
best_tour = [-1] * n

tour[0], visited[start_node] = start_node, True
tsp(graph, n, tour, visited, 1, min_cost, best_tour)

print("Best Tour:", " -> ".join(map(str, best_tour)) + f" -> {best_tour[0]}")
print("Minimum Cost:", min_cost[0])
```

```
main()
```

Output:

Best Tour: 0 -> 1 -> 2 -> 3 -> 0

Minimum Cost: 110