

FLAT-ASSIGNMENT-2

NAME:T.TARUNI

ROLL NO:21071A6758

Travelling Salesman Problem

Travelling Salesman Problem (TSP) is a classic combinatorics problem of theoretical computer science. The problem asks to find the shortest path in a graph with the condition of visiting all the nodes only one time and returning to the origin city.

Algorithm-

Algorithm: Traveling-Salesman-Problem

Cost (1, {}, 1) = 0

for s = 2 to n do

 for all subsets S belongs to {1, 2, 3, ..., n} of size s

 Cost (s, S, 1) = Infinity

 for all i ∈ S and i ≠ 1

 Cost (i, S, j) = min {Cost (i, S - {i}, j) + dist(i, j) for j ∈ S and i ≠ j}

Return min(i) Cost (i, {1, 2, 3, ..., n}, j) + d(j, i)

Code-

```
#include <bits/stdc++.h>
using namespace std;
#define V 4
#define MAX 1000000
int tsp(int graph[][V], int s)
{
    vector<int> vertex;
    for (int i = 0; i < V; i++)
        if (i != s)
            vertex.push_back(i);
    int min_cost = MAX;
    while(next_permutation(vertex.begin(), vertex.end()))
    {
        int current_cost = 0;
        int j = s;
        for (int i = 0; i < vertex.size(); i++) {
            current_cost += graph[j][vertex[i]];
            j = vertex[i];
        }
    }
}
```

```

        }
        current_cost += graph[j][s];
        min_cost = min(min_cost, current_cost);
        return min_cost
    }
}

int main()
{
    int graph[][V] = { { 0, 10, 15, 20 }, { 10, 0, 35, 25 }, { 15, 35, 0, 30 }, { 20, 25, 30, 0 } };
    int s = 0;
    cout << tsp(graph, s) << endl;
    return 0;
}

```

Output-

80

