

Name: Atharva Bilonikar

Uid No: 2021700011

CSE DS D1

DAA exp 9

Aim: Approximation algorithms (Travelling Salesman Problem)

Code:

```
#include <bits/stdc++.h>
using namespace std;
#define V 4

// implementation of traveling Salesman Problem
int travllingSalesmanProblem(int graph[][V], int s)
{
    // store all vertex apart from source vertex
    vector<int> vertex;
    for (int i = 0; i < V; i++)
        if (i != s)
            vertex.push_back(i);

    // store minimum weight Hamiltonian Cycle.
    int min_path = INT_MAX;
    do {

        // store current Path weight(cost)
        int current_pathweight = 0;

        // compute current path weight
        int k = s;
        for (int i = 0; i < vertex.size(); i++) {
            current_pathweight += graph[k][vertex[i]];
            k = vertex[i];
        }
        current_pathweight += graph[k][s];

        // update minimum
        min_path = min(min_path, current_pathweight);

    } while (
        next_permutation(vertex.begin(), vertex.end()));

    return min_path;
}
```

```

}

// Driver Code
int main()
{
    // matrix representation of graph
    int graph[][V] = { { 0, 10, 15, 20 },
                       { 10, 0, 35, 25 },
                       { 15, 35, 0, 30 },
                       { 20, 25, 30, 0 } };

    int s = 0;
    cout << travllingSalesmanProblem(graph, s) << endl;
    return 0;
}

```

Output:



```

OUTPUT  TERMINAL  2  JUPYTER
TERMINAL
PS C:\Users\Shreya\Desktop\code> cd "C:\Users\Shreya\Desktop\code\" ; if ($?) { g++ daa6.cpp -o daa6 } ; if ($?) { .\daa6 }
80
PS C:\Users\Shreya\Desktop\code>

```

Conclusion : In this experiment I understood how to implement travelling salesman algorithm using approximation algorithm .