#### Implement Travelling Salesman Problem using Branch and Bound.

#### Code:

#include<iostream>

using namespace std;

#define INT\_MAX 999

#define V 4

int n=4;

int VISITED\_ALL = (1<<n) -1;

int dp[16][4];

int  tsp(int mask,int pos,int dist[][V]){

    if(mask==VISITED\_ALL){

        return dist[pos][0];

    }

    if(dp[mask][pos]!=-1){

       return dp[mask][pos];

    }

    int ans = INT\_MAX;

    for(int city=0;city<n;city++){

        if((mask&(1<<city))==0){

            int newAns = dist[pos][city] + tsp( mask|(1<<city), city,dist);

            ans = min(ans, newAns);

        }

    }

    return dp[mask][pos] = ans;

}

int main(){

    int g[V][V];

    cout<<"Enter adjacency matrix\nEnter 999 if no edges between vertices\n";

    for (int i=0;i<V;i++)

    {

        for(int j=0;j<V;j++)

        {

            cin>>g[i][j];

        }

    }

    for(int i=0;i<(1<<n);i++){

        for(int j=0;j<n;j++){

            dp[i][j] = -1;

        }

    }

    cout<<"Travelling Saleman Distance is "<<tsp(1,0,g);

return 0;

}

