|  |  |
| --- | --- |
| **Course Code: CT2352** | **Course Name: Lab-DAA** |

|  |  |
| --- | --- |
| **Name of Students: S Akshansh** | **Semester/ Section: 6 A** |
| **Roll No: 72** | **Enroll No: 19010927** |

**Practical No 7**

|  |
| --- |
| **Aim**: Given a set of cities and distance between every pair of cities. Write a program is to find the shortest possible route that visits every city exactly once and returns to the starting point. Also analyse its time complexity. |
| **Theory:-**  The traveling salesman problems abide by a salesman and a set of cities. The salesman has to visit every one of the cities starting from a certain one (e.g., the hometown) and to return to the same city. The challenge of the problem is that the traveling salesman needs to minimize the total length of the trip.  Suppose the cities are x1 x2..... xn where cost cij denotes the cost of travelling from city xi to xj. The travelling salesperson problem is to find a route starting and ending at x1 that will take in all cities with the minimum cost.  **Example:** A newspaper agent daily drops the newspaper to the area assigned in such a manner that he has to cover all the houses in the respective area with minimum travel cost. Compute the minimum travel cost.  The area assigned to the agent where he has to drop the newspaper :  Travelling Sales Person Problem  Solution: The cost- adjacency matrix of graph G is as follows:  costij =  Travelling Sales Person Problem  The tour starts from area H1 and then select the minimum cost area reachable from H1.  Travelling Sales Person Problem  Mark area H6 because it is the minimum cost area reachable from H1 and then select minimum cost area reachable from H6.  Travelling Sales Person Problem  Mark area H7 because it is the minimum cost area reachable from H6 and then select minimum cost area reachable from H7.  Travelling Sales Person Problem  Mark area H8 because it is the minimum cost area reachable from H8.  Travelling Sales Person Problem  Mark area H5 because it is the minimum cost area reachable from H5.  Travelling Sales Person Problem  Mark area H2 because it is the minimum cost area reachable from H2.  Travelling Sales Person Problem  Mark area H3 because it is the minimum cost area reachable from H3.  Travelling Sales Person Problem  Mark area H4 and then select the minimum cost area reachable from H4 it is H1.So, using the greedy strategy, we get the following.  4 3 2 4 3 2 1 6  H1 → H6 → H7 → H8 → H5 → H2 → H3 → H4 → H1.  Thus the minimum travel cost = 4 + 3 + 2 + 4 + 3 + 2 + 1 + 6 = 25  **ALGORITHM:**  s: starting point;  N: a subset of input cities;  dist(): distance among the cities  Cost : TSP result  Visited[N] = 0;  Cost = 0;  Procedure TSP(N,s)  Visited[s] = 1;  if|N| = 2 and k != s then  Cost(N,k) = dist(s,k);  Return Cost;  else  for j ∈ N do  for i ∈ N and visited[i] = 0 do  if j != I and j != s then  Cost(N,j) = min (TSP(N-{i}, j + dist(j,i))  Visited[j] = 1;  end  end  end  end  Return Cost;  end |
| **Code:-**  #include<iostream>  using namespace std;  int ary[10][10],completed[10],n,cost=0;  void takeInput()  {  int i,j;  cout<<"\nEnter the number of Cities: ";  cin>>n;  cout<<"\nEnter the Cost Matrix\n";  for(i=0;i < n;i++)  {  cout<<"\nEnter Elements of Row: "<<i+1<<"\n";  for( j=0;j < n;j++)  cin>>ary[i][j];  completed[i]=0;  }  cout<<"\n\nThe cost list is:";  for( i=0;i < n;i++)  {  cout<<"\n";  for(j=0;j < n;j++)  cout<<"\t"<<ary[i][j];  }  }  int least(int c)  {  int i,nc=999;  int min=999,kmin;  for(i=0;i < n;i++)  {  if((ary[c][i]!=0)&&(completed[i]==0))  if(ary[c][i]+ary[i][c] < min)  {  min=ary[i][0]+ary[c][i];  kmin=ary[c][i];  nc=i;  }  }  if(min!=999)  cost+=kmin;  return nc;  }  void mincost(int city)  {  int i,ncity;  completed[city]=1;  cout<<city+1<<"--->";  ncity=least(city);  if(ncity==999)  {  ncity=0;  cout<<ncity+1;  cost+=ary[city][ncity];  return;  }  mincost(ncity);  }  int main()  {  takeInput();  cout<<"\n\nThe Path is:\n";  mincost(0); //passing 0 because starting vertex  cout<<"\n\nMinimum cost is "<<cost;  return 0;  } |
| **Output:-** |
| **Conclusion:** Thus, I have executed the code for travelling salesforce problem . Time complexity is O(n2 \* 2n). |