TRAVELLING SALESMAN PROBLEM USING

BRANCH AND BOUND

import java.util.\*;

class Tsp

{

int cost,n;

int [][]reduction(int temp\_cost[][],int n)

{

int min;

cost=0;

for(int i=0;i<n;i++)

{

min=1000;

for(int j=0;j<n;j++)

{

if(min>temp\_cost[i][j])

min=temp\_cost[i][j];

}

if(min==1000)

min=0;

for(int j=0;j<n;j++)

{

temp\_cost[i][j]-=min;

}

cost+=min;

}

for(int i=0;i<n;i++)

{

min=1000;

for(int j=0;j<n;j++)

{

if(min>temp\_cost[j][i])

min=temp\_cost[j][i];

}

if(min==1000)

min=0;

for(int j=0;j<n;j++)

{

temp\_cost[j][i]-=min;

}

cost+=min;

}

return temp\_cost;

}

int tsp\_path(int temp\_cost[][],int curr\_path[],int pos,int n)

{

int flag,curr\_cost=cost,mincost=1000,ver=0;

int mat1[][]=new int[10][10];

int mat2[][] = new int[10][10];

while(pos<n-1)

{

int k=curr\_path[pos];

mincost=1000;

for(int i=0;i<n;i++)

{

cost=0;

int temp=temp\_cost[k][i];

flag=0;

for(int j=0;j<=pos;j++)

{

if(i==curr\_path[j])

{

flag=1;

break;

}

}

if(flag==1)

continue;

for(int j=0;j<n;j++)

{

for(int l=0;l<n;l++)

{

if(j==k)

mat1[j][l]=9999;

else if(l==i)

mat1[j][l]=9999;

else if(j==i&&l==k)

mat1[j][l]=9999;

else

mat1[j][l]=temp\_cost[j][l];

}

}

mat1[i][0]=9999;

mat1=reduction(mat1,n);

cost+=curr\_cost+temp;

if(mincost>cost)

{

for(int w=0;w<n;w++)

{

for(int p=0;p<n;p++)

{

mat2[w][p]=mat1[w][p];

}

}

ver=i;

mincost=cost;

}

}

for(int w=0;w<n;w++)

{

for(int p=0;p<n;p++)

{

temp\_cost[w][p]=mat2[w][p];

}

}

curr\_cost=mincost;

curr\_path[++pos]=ver;

}

System.out.print("Path of Travel: ");

for(int i=0;i<pos;i++)

System.out.print(curr\_path[i]+"->");

System.out.print(curr\_path[pos]);

return curr\_cost;

}

public static void main(String a[])

{

int n,pos=-1;

int cost\_mat[][]=new int[10][10];

int temp\_cost[][]=new int[10][10];

int curr\_path[]=new int[10];

Scanner s= new Scanner(System.in);

System.out.println("Travelling Salesman Problem");

System.out.println("Enter the number of places to visit");

n=s.nextInt();

System.out.println("Enter the cost matrix\nEnter 9999 for INFINITY");

for(int i=0;i<n;i++)

{

for(int j=0;j<n;j++)

{

cost\_mat[i][j]=s.nextInt();

temp\_cost[i][j]=cost\_mat[i][j];

}

}

Tsp obj = new Tsp();

temp\_cost=obj.reduction(temp\_cost,n);

curr\_path[++pos]=0;

obj.cost = obj.tsp\_path(temp\_cost,curr\_path,pos,n);

System.out.println("\nCost of Travel: " + obj.cost);

}

}

/\*

9999 20 30 10 11

15 9999 16 4 2

3 5 9999 2 4

19 6 18 9999 3

16 4 7 16 9999

\*/

OUTPUT:

