LAB 10| Artificial Intelligence

Aim: Travelling salesman problem using nearest neighbour heuristic and greedy edge heuristic.

- Using Nearest Neighbour Heuristic:

Code :

```
n = int(input("Enter No. of Nodes : "))

print("Enter Adjacency Matrix : ")
arr = [[0 for j in range(n)] for i in range(n)]
for i in range(0,n):
    arr[i] = list(map(int,input().split()))

print("Adjacency Matrix is : ")
print(arr)

for p in range(n):
    visited = [0 for l in range(n)]
    start=p
    cost=0
    j=1
    curr=start
    visited[start]=1
    print("Path For City " + str(curr+1))
    print(curr+1,end=' ')
    while j<n:
        nearest = arr[curr][curr]
        k = curr
        for i in range(n):
            if arr[curr][i]<nearest and visited[i]==0:
                nearest=arr[curr][i]
                k=i
        curr=k
        cost=cost+nearest
        visited[k]=1
        j=j+1
        print("> " + str(k+1),end=' ')
    cost=cost+arr[curr][start]
    print("")
    print("Cost for City " + str(p+1) + "=" + str(cost))
    print("-----------------------------------")
```

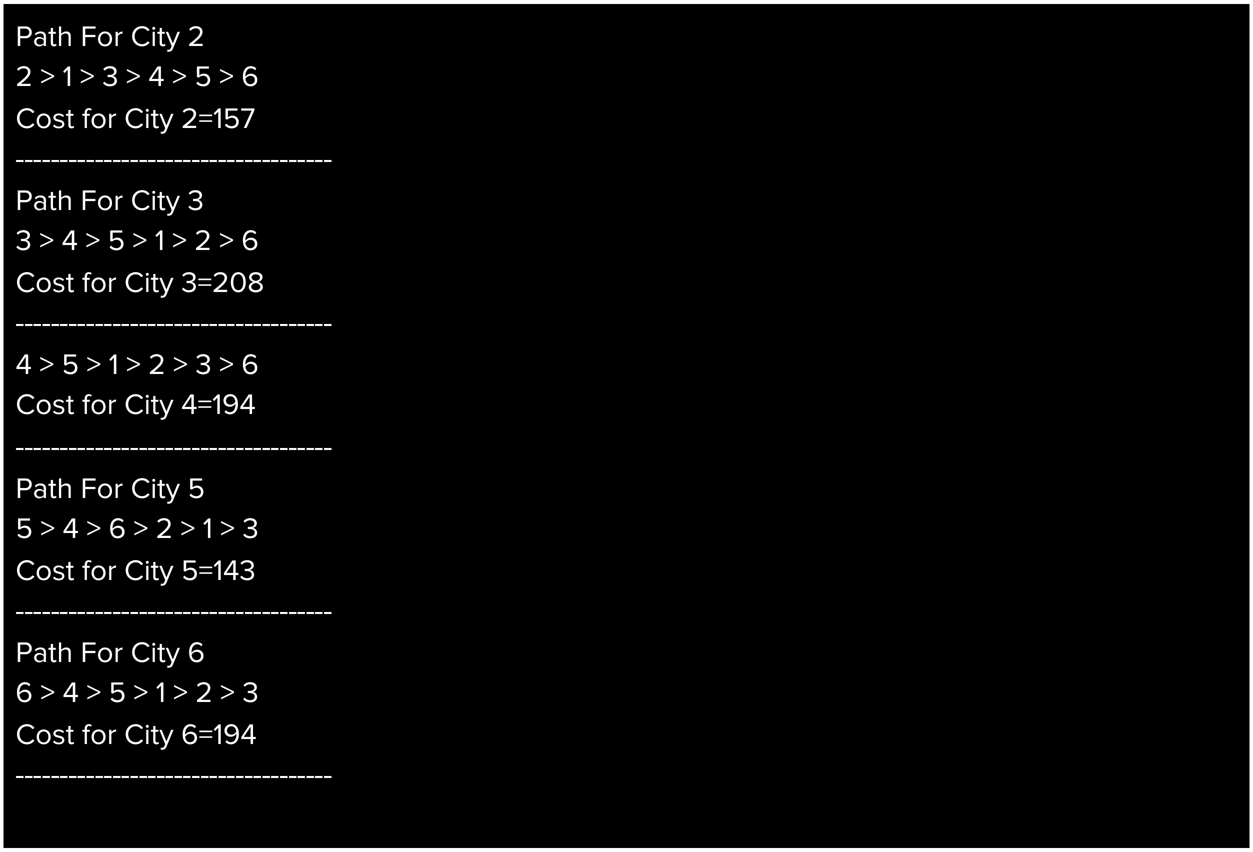
Output :

```
Enter No. of Nodes : 6

Enter Adjacency Matrix :
999 10 20 30 40 50
10 999 31 21 51 41
20 31 999 12 59 100
30 21 12 999 5 8
40 51 59 5 999 69
50 41 100 8 69 999

Adjacency Matrix is :
[[999, 10, 20, 30, 40, 50],
 [10, 999, 31, 21, 51, 41],
 [20, 31, 999, 12, 59, 100],
[30, 21, 12, 999, 5, 8],
[40, 51, 59, 5, 999, 69],
[50, 41, 100, 8, 69, 999]]

Path For City 1
1 > 2 > 4 > 5 > 3 > 6
Cost for City 1=245
-----------------------------------
```



```
Path For City 2
2 > 1 > 3 > 4 > 5 > 6
Cost for City 2=157
------------------------------------
Path For City 3
3 > 4 > 5 > 1 > 2 > 6
Cost for City 3=208
------------------------------------
4 > 5 > 1 > 2 > 3 > 6
Cost for City 4=194
------------------------------------
Path For City 5
5 > 4 > 6 > 2 > 1 > 3
Cost for City 5=143
------------------------------------
Path For City 6
6 > 4 > 5 > 1 > 2 > 3
Cost for City 6=194
------------------------------------
```

- Using Greedy Edge Heuristic :

Code:

```
n = int(input("Enter No. of Nodes : "))

print("Enter Adjacency Matrix : ")
arr = [[0 for j in range(n)] for i in range(n)]
for i in range(0,n):
   arr[i] = list(map(int,input().split()))

print("Adjacency Matrix is : ")
print(arr)
```

```
start=0
end=0
visited = [0 for l in range(n)]
cost=0
min=9999
for i in range(n):
    for j in range(n):
        if arr[i][j]<min:
            start=i
            end=j
            min=arr[i][j]
            cost=arr[i][j]

print(str(start+1) + " " + str(end+1))
c=2
visited[start]=1
visited[end]=1
while c<n:
    min_edge=9999
    node=0
    for j in range(n):
        if arr[start][j]<min_edge and visited[j]==0:
            node=j
            min_edge=arr[start][j]

    k=9999
    p=0
    for j in range(n):
        if arr[end][j]<k and visited[j]==0:
            p=j
            k=arr[end][j]
    if k<min_edge:
        print(str(end+1)+" " + str(p+1))
        cost=cost+k
        visited[p]=1
        end=p
    else:
        cost=cost+min_edge
        print(str(start+1)+" "+str(node+1))
        start=node
        visited[node]=1
    c=c+1

print(str(start+1)+" "+str(end+1))
cost=cost+arr[start][end]
print("Cost =",cost)
```

Output:

```
Enter No. of Nodes : 6

Enter Adjacency Matrix :
999 10 20 30 40 50
10 999 31 21 51 41
20 31 999 12 59 100
30 21 12 999 5 8
40 51 59 5 999 69
50 41 100 8 69 999

Adjacency Matrix is :
[[999, 10, 20, 30, 40, 50],
 [10, 999, 31, 21, 51, 41],
 [20, 31, 999, 12, 59, 100],
[30, 21, 12, 999, 5, 8],
 [40, 51, 59, 5, 999, 69],
 [50, 41, 100, 8, 69, 999]]

4 5
4 6
5 1
```

```
1 2
2 3
6 3
Cost = 194
```