

LAB 2 – TRAVELLING SALESMAN PROBLEM

AI LAB

RAJAT KUMAR

RA1911003010652

ALGORITHM:

Step 1: Consider city 1 as the starting and ending point.

Step 2: Generate all $(n-1)!$ Permutations of cities.

Step 3: Calculate cost of every permutation and keep track of minimum cost permutation.

Step 4: Return the permutation with minimum cost.

CODE:

```
from sys import maxsize

from itertools import permutations

V = 4

# implementation of traveling Salesman Problem

def travellingSalesmanProblem(graph, s):

    # store all vertex apart from source vertex
    vertex = []

    for i in range(V):
        if i != s:
```

```
vertex.append(i)
```

```
# store minimum weight Hamiltonian Cycle
```

```
min_path = maxsize
```

```
next_permutation=permutations(vertex)
```

```
for i in next_permutation:
```

```
    # store current Path weight(cost)
```

```
    current_pathweight = 0
```

```
    # compute current path weight
```

```
    k = s
```

```
    for j in i:
```

```
        current_pathweight += graph[k][j]
```

```
        k = j
```

```
    current_pathweight += graph[k][s]
```

```
    # update minimum
```

```
    min_path = min(min_path, current_pathweight)
```

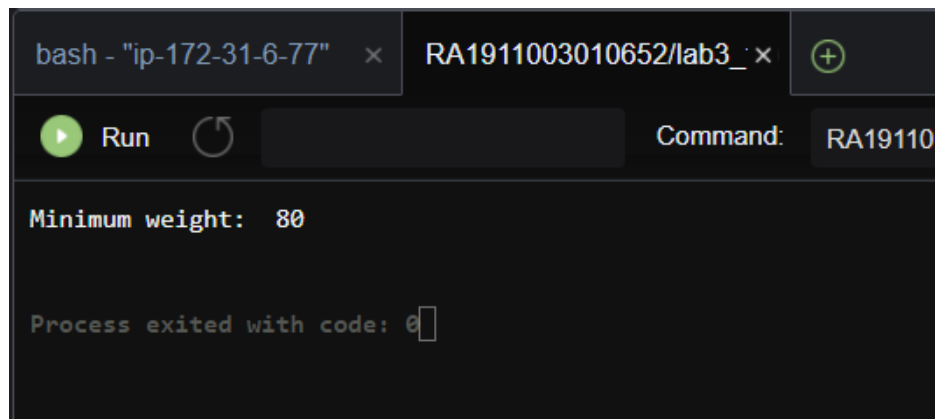
```
return min_path
```

```
# Driver Code
```

```
if __name__ == "__main__":
```

```
# matrix representation of graph
graph = [[0, 10, 15, 20], [10, 0, 35, 25],
         [15, 35, 0, 30], [20, 25, 30, 0]]
s = 0
print("Minimum weight: ",travellingSalesmanProblem(graph, s))
```

OUTPUT:



The screenshot shows a terminal window with a dark background. At the top, there are two tabs: "bash - 'ip-172-31-6-77'" and "RA1911003010652/lab3_". Below the tabs, there is a "Run" button with a green play icon and a "Command:" field containing "RA19110". The main area of the terminal displays the output of the program: "Minimum weight: 80" and "Process exited with code: 0".

```
bash - "ip-172-31-6-77" × RA1911003010652/lab3_ × (+)
Run Command: RA19110
Minimum weight: 80
Process exited with code: 0
```

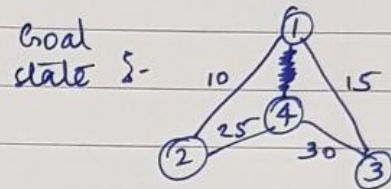
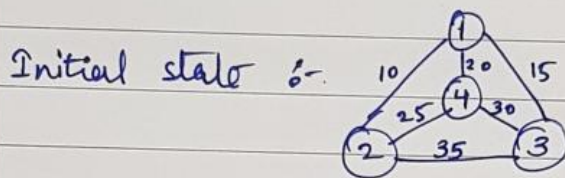
RESULT: Hence, the implementation of Travelling Salesman Person was successfully done.

AI LAB-2

AIM: Developing agent programs for real world problems
Travelling Salesman Problem (TSP).

PROBLEM FORMULATION:

For a given complete graph with n vertices and weight function defined on the edges, the objective is to construct a tour.



PROBLEM SOLVING :

We start at vertex 1 and find the minimum cost path with 1 as starting point, as ending point and all vertices appearing exactly once.

for path $1 \rightarrow 2$ the minimum cost would be through direct path

