INSTITUTE OF COMPUTER TECHNOLOGY

B-TECH COMPUTER SCIENCE ENGINEERING 2025-26

SUBJECT:-ALGORITHM ANALYSIS & DESIGN

NAME: Rahul Prajapati

ENRLL. NO: 23162171020

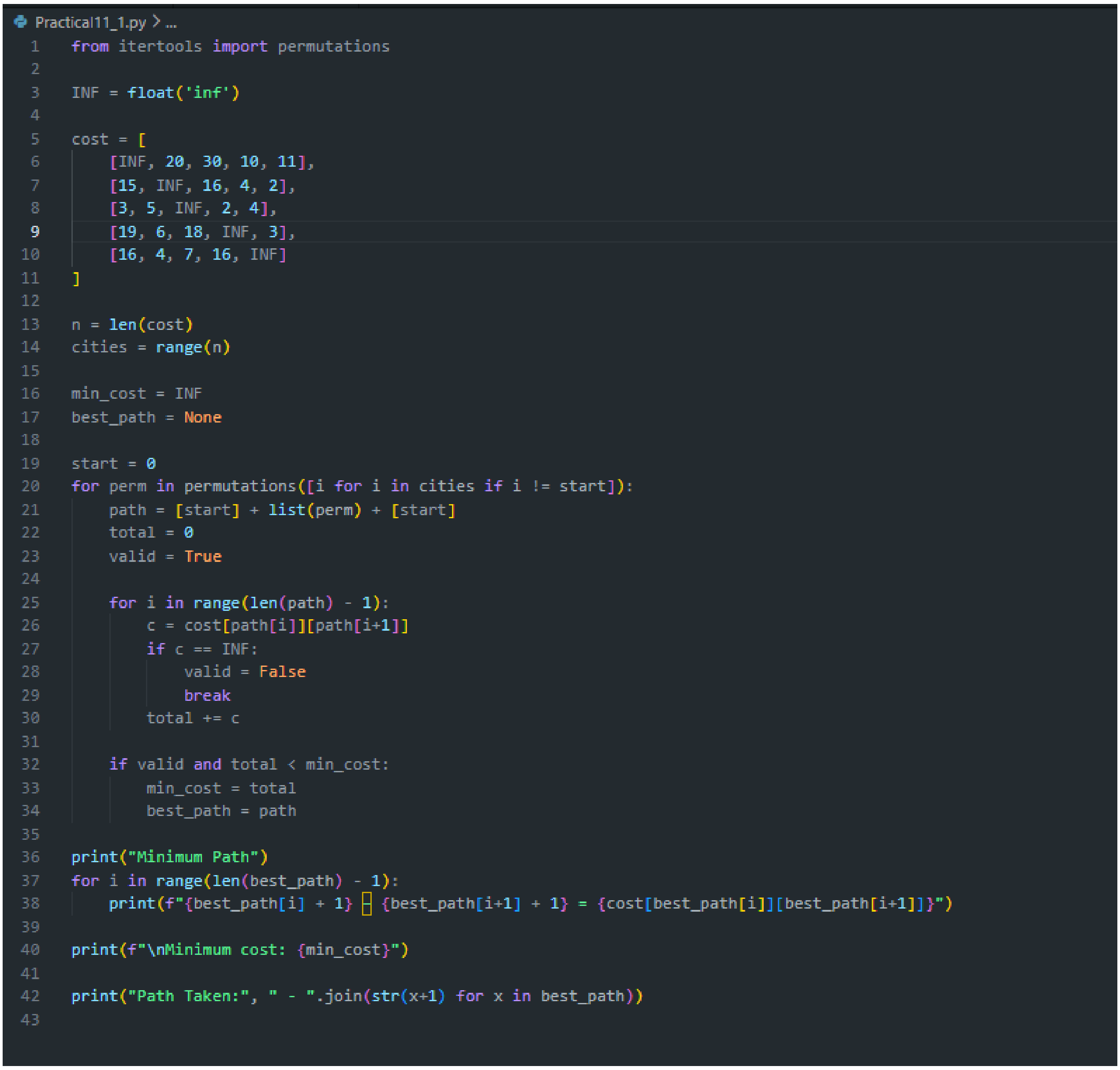
BRANCH: CYBER SECURITY

BATCH: 52

PRACTICAL_11

AIM: "Rocket Singh: Salesman of the Year" is a travelling salesman, who sales good in various cities. One day in the morning, he decided to visit all the cities to sales good and come back to the starting city (from where he has started). Travelling Salesman Problem (TSP) is a touring problem in which n cities and distance between each pair is given. We have to help him to find a shortest route to visit each city exactly once and come back to the starting point.

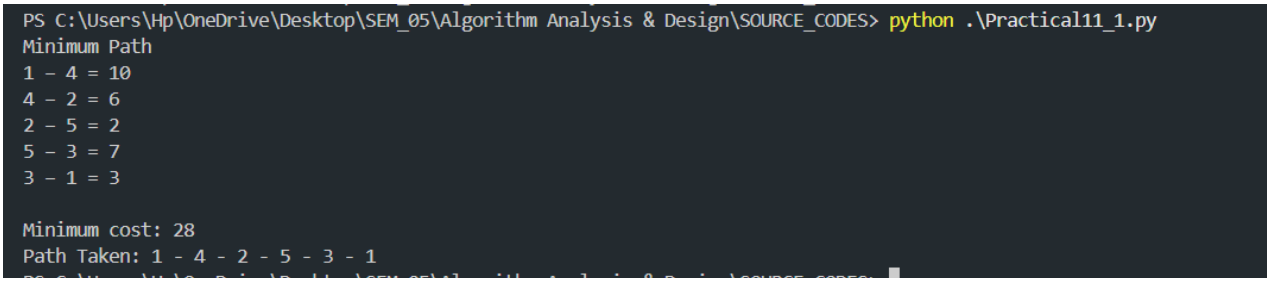
CODE:



Practical11_1.py > ...

```python
from itertools import permutations

INF = float('inf')

cost = [
    [INF, 20, 30, 10, 11],
    [15, INF, 16, 4, 2],
    [3, 5, INF, 2, 4],
    [19, 6, 18, INF, 3],
    [16, 4, 7, 16, INF]
]

n = len(cost)
cities = range(n)

min_cost = INF
best_path = None

start = 0
for perm in permutations([i for i in cities if i != start]):
    path = [start] + list(perm) + [start]
    total = 0
    valid = True

    for i in range(len(path) - 1):
        c = cost[path[i]][path[i+1]]
        if c == INF:
            valid = False
            break
        total += c

    if valid and total < min_cost:
        min_cost = total
        best_path = path

print("Minimum Path")
for i in range(len(best_path) - 1):
    print(f"{best_path[i] + 1} – {best_path[i+1] + 1} = {cost[best_path[i]][best_path[i+1]]}")

print(f"\nMinimum cost: {min_cost}")

print("Path Taken:", " - ".join(str(x+1) for x in best_path))

```

OUTPUT:



```
PS C:\Users\Hp\OneDrive\Desktop\SEM_05\Algorithm Analysis & Design\SOURCE_CODES> python .\Practical11_1.py
Minimum Path
1 – 4 = 10
4 – 2 = 6
2 – 5 = 2
5 – 3 = 7
3 – 1 = 3

Minimum cost: 28
Path Taken: 1 - 4 - 2 - 5 - 3 - 1
PS C:\Users\Hp\OneDrive\Desktop\SEM_05\Algorithm Analysis & Design\SOURCE_CODES>
```