DAA Practical 7

Name: Mohammad Anas Ajaz
Class: A4(B4)
Roll No. 61

Aim:
Implement Hamiltonian Cycle using Backtracking.

Problem Statement:
The Smart City Transportation Department is designing a night-patrol route for
security vehicles.
Each area of the city is represented as a vertex in a graph, and a road between two
areas is represented as an edge.
The goal is to find a route that starts from the main headquarters (Area A), visits
each area exactly once, and returns back to the headquarters — forming a Hamiltonian Cycle.
If such a route is not possible, display a suitable message.

Code:

```
def checking(v, graph, path, pos):
    if graph[path[pos - 1]][v] == 0:
        return False
    if v in path:
        return False
    return True

def hamiltonian_cycle_util(graph, path, pos):
    n = len(graph)
    if pos == n:
        if graph[path[pos - 1]][path[0]] == 1:
            return True
        return False

    for v in range(1, n):
```

```
        if checking(v, graph, path, pos):
            path[pos] = v
            if hamiltonian_cycle_util(graph, path, pos + 1):
                return True
            path[pos] = -1
    return False

def hamiltonian_cycle(graph, vertices):
    n = len(graph)
    path = [-1] * n
    path[0] = 0

    if not hamiltonian_cycle_util(graph, path, 1):
        print("No Hamiltonian Cycle exists")
        return

    print("Hamiltonian Cycle exists:")
    for vertex in path:
        print(vertices[vertex], end=" -> ")
    print(vertices[path[0]])

graph1 = [
    [0, 1, 1, 0, 1],
    [1, 0, 1, 1, 0],
    [1, 1, 0, 1, 0],
    [0, 1, 1, 0, 1],
    [1, 0, 0, 1, 0]
]
vertices1 = ['A', 'B', 'C', 'D', 'E']
print("Example 1:")
hamiltonian_cycle(graph1, vertices1)
print()

graph2 = [
    [0, 1, 1, 0, 1],
    [1, 0, 1, 1, 0],
    [1, 1, 0, 1, 1],
    [0, 1, 1, 0, 1],
    [1, 0, 1, 1, 0]
]
```

```
vertices2 = ['T', 'M', 'S', 'H', 'C']
print("Example 2:")
hamiltonian_cycle(graph2, vertices2)
```

Output:

```
Example 1:
Hamiltonian Cycle exists:
A -> B -> C -> D -> E -> A

Example 2:
Hamiltonian Cycle exists:
T -> M -> S -> H -> C -> T
```

GFG Competitive Coding :