Practical-7

Name: Krish Parothi

Section: A4

Batch: B3

Roll Number: 49

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.

1) Adjacency Matrix

A B C D E

A 0 1 1 0 1

B 1 0 1 1 0

C 1 1 0 1 0

D 0 1 1 0 1

E 1 0 0 1 0

1) Adjacency Matrix

T M S H C

T 0 1 1 0 1

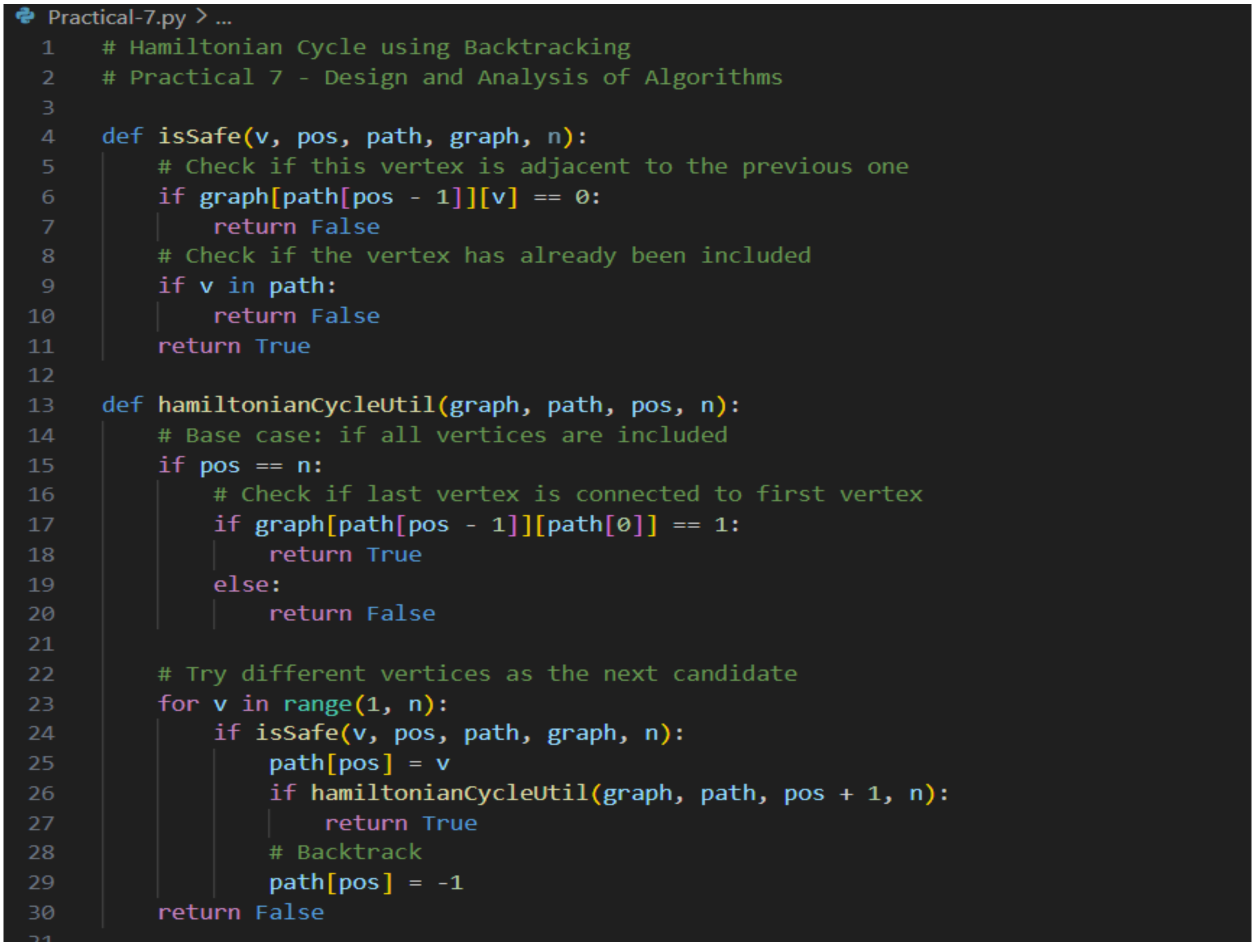
M 1 0 1 1 0

S 1 1 0 1 1

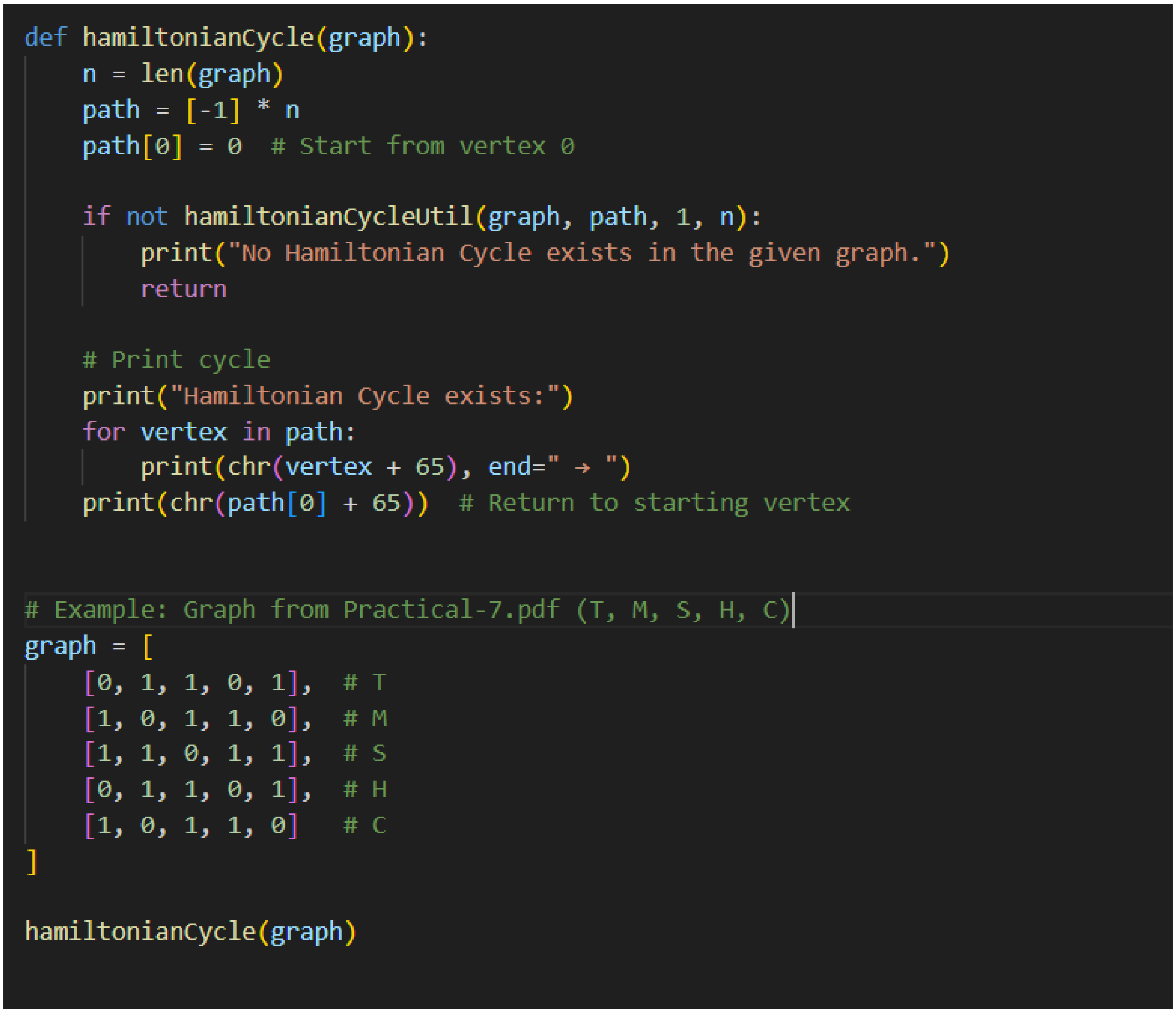
H 0 1 1 0 1

C 1 0 1 1 0

CODE:



```python
# Hamiltonian Cycle using Backtracking
# Practical 7 - Design and Analysis of Algorithms

def isSafe(v, pos, path, graph, n):
    # Check if this vertex is adjacent to the previous one
    if graph[path[pos - 1]][v] == 0:
        return False
    # Check if the vertex has already been included
    if v in path:
        return False
    return True

def hamiltonianCycleUtil(graph, path, pos, n):
    # Base case: if all vertices are included
    if pos == n:
        # Check if last vertex is connected to first vertex
        if graph[path[pos - 1]][path[0]] == 1:
            return True
        else:
            return False

    # Try different vertices as the next candidate
    for v in range(1, n):
        if isSafe(v, pos, path, graph, n):
            path[pos] = v
            if hamiltonianCycleUtil(graph, path, pos + 1, n):
                return True
            # Backtrack
            path[pos] = -1
    return False
```

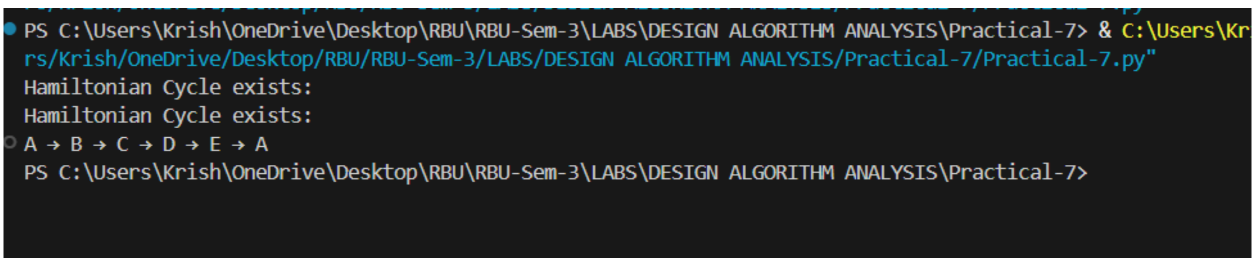


```python
def hamiltonianCycle(graph):
    n = len(graph)
    path = [-1] * n
    path[0] = 0  # Start from vertex 0

    if not hamiltonianCycleUtil(graph, path, 1, n):
        print("No Hamiltonian Cycle exists in the given graph.")
        return

    # Print cycle
    print("Hamiltonian Cycle exists:")
    for vertex in path:
        print(chr(vertex + 65), end=" → ")
    print(chr(path[0] + 65))  # Return to starting vertex


# Example: Graph from Practical-7.pdf (T, M, S, H, C)
graph = [
    [0, 1, 1, 0, 1],  # T
    [1, 0, 1, 1, 0],  # M
    [1, 1, 0, 1, 1],  # S
    [0, 1, 1, 0, 1],  # H
    [1, 0, 1, 1, 0]   # C
]

hamiltonianCycle(graph)
```

Code in Text:

```python
# Hamiltonian Cycle using Backtracking
# Practical 7 - Design and Analysis of Algorithms

def isSafe(v, pos, path, graph, n):
    # Check if this vertex is adjacent to the previous one
    if graph[path[pos - 1]][v] == 0:
        return False
    # Check if the vertex has already been included
    if v in path:
        return False
    return True

def hamiltonianCycleUtil(graph, path, pos, n):
    # Base case: if all vertices are included
    if pos == n:
        # Check if last vertex is connected to first vertex
        if graph[path[pos - 1]][path[0]] == 1:
            return True
        else:
            return False

    # Try different vertices as the next candidate
    for v in range(1, n):
        if isSafe(v, pos, path, graph, n):
            path[pos] = v
            if hamiltonianCycleUtil(graph, path, pos + 1, n):
                return True
            # Backtrack
            path[pos] = -1
    return False

def hamiltonianCycle(graph):
    n = len(graph)
    path = [-1] * n
    path[0] = 0  # Start from vertex 0

    if not hamiltonianCycleUtil(graph, path, 1, n):
        print("No Hamiltonian Cycle exists in the given graph.")
        return

    # Print cycle
    print("Hamiltonian Cycle exists:")
    for vertex in path:
        print(chr(vertex + 65), end=" → ")
    print(chr(path[0] + 65))  # Return to starting vertex


# Example: Graph from Practical-7.pdf (T, M, S, H, C)
graph = [
    [0, 1, 1, 0, 1],  # T
    [1, 0, 1, 1, 0],  # M
    [1, 1, 0, 1, 1],  # S
    [0, 1, 1, 0, 1],  # H
    [1, 0, 1, 1, 0]   # C
]

hamiltonianCycle(graph)
```

OUTPUT:



```
PS C:\Users\Krish\OneDrive\Desktop\RBU\RBU-Sem-3\LABS\DESIGN ALGORITHM ANALYSIS\Practical-7> & C:\Users\Kr
rs/Krish/OneDrive/Desktop/RBU/RBU-Sem-3/LABS/DESIGN ALGORITHM ANALYSIS/Practical-7/Practical-7.py"
Hamiltonian Cycle exists:
Hamiltonian Cycle exists:
A → B → C → D → E → A
PS C:\Users\Krish\OneDrive\Desktop\RBU\RBU-Sem-3\LABS\DESIGN ALGORITHM ANALYSIS\Practical-7>
```