VIT-AP
UNIVERSITY

LAB REPORT

Assignment 6

NAME : P.SATHWIKA

REGISTRATION NUMBER : 21BCE8118

SUBJECT : Design and Analysis of Algorithms.

LAB SLOT : L53+L54

IMPLEMENTATION OF HAMILTON CYCLE

USING GRAPHS

Hamiltonian Cycle or Circuit in a graph G is a cycle that visits every vertex of G exactly once and returns to the starting vertex.

ALGORITHM:

Create a function hamiltonian_cycle(graph) that takes a graph as input.

Initialize a list path to store the Hamiltonian cycle path.

Create a helper function is_valid(vertex, pos, path) to check if adding vertex at position pos in the path is valid.

In the main function:

Start from any vertex as the current vertex (let’s say vertex 0).

Add this vertex to the path.

Call the recursive function hamiltonian_util(graph, path).

In the recursive function:

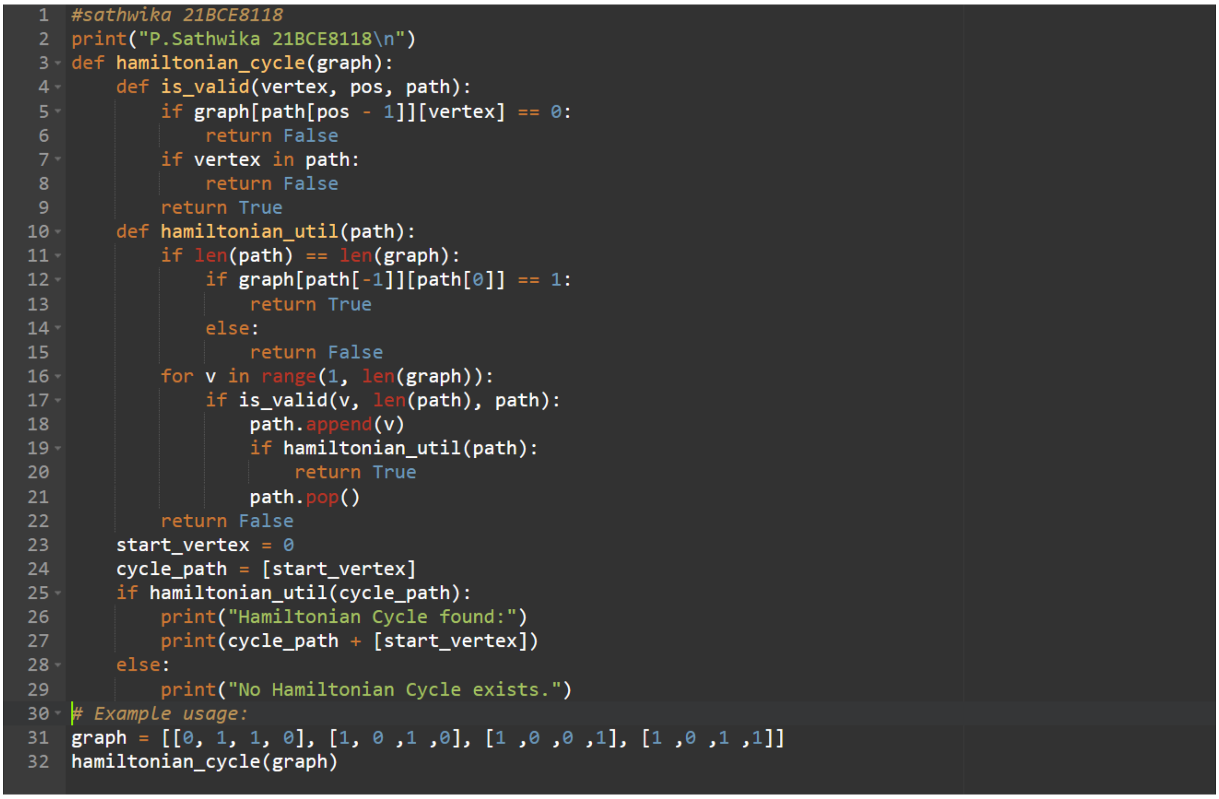
If all vertices are included in the path, check if there is an edge from the last included vertex to the starting vertex (0).

If yes, return True (Hamiltonian cycle found).

Otherwise, backtrack and try other vertices.

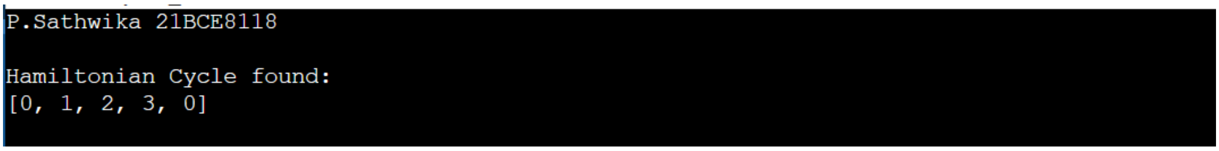
CODE:

```
#sathwika 21BCE8118

print("P.Sathwika 21BCE8118\n")

def hamiltonian_cycle(graph):

    def is_valid(vertex, pos, path):

        if graph[path[pos - 1]][vertex] == 0:

            return False

        if vertex in path:
            return False
        return True
    def hamiltonian_util(path):
        if len(path) == len(graph):
            if graph[path[-1]][path[0]] == 1:
                return True
            else:
                return False
        for v in range(1, len(graph)):
            if is_valid(v, len(path), path):
                path.append(v)
                if hamiltonian_util(path):
                    return True
                path.pop()

        return False
    start_vertex = 0
    cycle_path = [start_vertex]
    if hamiltonian_util(cycle_path):
        print("Hamiltonian Cycle found:")
        print(cycle_path + [start_vertex])
    else:
        print("No Hamiltonian Cycle exists.")
# Example usage:
graph = [[0, 1, 1, 0], [1, 0 ,1 ,0], [1 ,0 ,0 ,1], [1 ,0 ,1 ,1]]
hamiltonian_cycle(graph)
```



```python
#sathwika 21BCE8118
print("P.Sathwika 21BCE8118\n")
def hamiltonian_cycle(graph):
    def is_valid(vertex, pos, path):
        if graph[path[pos - 1]][vertex] == 0:
            return False
        if vertex in path:
            return False
        return True
    def hamiltonian_util(path):
        if len(path) == len(graph):
            if graph[path[-1]][path[0]] == 1:
                return True
            else:
                return False
        for v in range(1, len(graph)):
            if is_valid(v, len(path), path):
                path.append(v)
                if hamiltonian_util(path):
                    return True
                path.pop()
        return False
    start_vertex = 0
    cycle_path = [start_vertex]
    if hamiltonian_util(cycle_path):
        print("Hamiltonian Cycle found:")
        print(cycle_path + [start_vertex])
    else:
        print("No Hamiltonian Cycle exists.")
# Example usage:
graph = [[0, 1, 1, 0], [1, 0 ,1 ,0], [1 ,0 ,0 ,1], [1 ,0 ,1 ,1]]
hamiltonian_cycle(graph)
```

OUTPUT:



```
P.Sathwika 21BCE8118

Hamiltonian Cycle found:
[0, 1, 2, 3, 0]
```