



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)

```

```

1 #sathwika 21BCE8118
2 print("P.Sathwika 21BCE8118\n")
3 def hamiltonian_cycle(graph):
4     def is_valid(vertex, pos, path):
5         if graph[path[pos - 1]][vertex] == 0:
6             return False
7         if vertex in path:
8             return False
9         return True
10    def hamiltonian_util(path):
11        if len(path) == len(graph):
12            if graph[path[-1]][path[0]] == 1:
13                return True
14            else:
15                return False
16        for v in range(1, len(graph)):
17            if is_valid(v, len(path), path):
18                path.append(v)
19                if hamiltonian_util(path):
20                    return True
21                path.pop()
22        return False
23    start_vertex = 0
24    cycle_path = [start_vertex]
25    if hamiltonian_util(cycle_path):
26        print("Hamiltonian Cycle found:")
27        print(cycle_path + [start_vertex])
28    else:
29        print("No Hamiltonian Cycle exists.")
30    # Example usage:
31    graph = [[0, 1, 1, 0], [1, 0, 1, 0], [1, 0, 0, 1], [1, 0, 1, 1]]
32    hamiltonian_cycle(graph)

```

OUTPUT:

```

P.Sathwika 21BCE8118

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

```