119. Hamiltonian Cycle Problem

Code:

```
def is_valid(v, pos, path, graph):

    if graph[path[pos - 1]][v] == 0:
        return False



    if v in path:
        return False


    return True

def ham_cycle_util(graph, path, pos):

    if pos == len(graph):


        if graph[path[pos - 1]][path[0]] == 1:
            return True
        else:
            return False

    for v in range(1, len(graph)):
        if is_valid(v, pos, path, graph):
            path[pos] = v



            if ham_cycle_util(graph, path, pos + 1):
                return True
```

```
            path[pos] = -1

    return False

def ham_cycle(graph):
    path = [-1] * len(graph)
    path[0] = 0

    if not ham_cycle_util(graph, path, 1):
        return None
    else:
        return path

graph = [
    [0, 1, 0, 1, 0],
    [1, 0, 1, 1, 1],
    [0, 1, 0, 0, 1],
    [1, 1, 0, 0, 1],
    [0, 1, 1, 1, 0]
]

result = ham_cycle(graph)
if result:
    print("Hamiltonian Cycle exists:")
    print(result + [result[0]])
else:
    print("No Hamiltonian Cycle exists")
```

output:

```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Documents/OriginLab/problem.py
Hamiltonian Cycle exists:
[0, 1, 2, 4, 3, 0]
PS C:\Users\karth>
```

time complexity:

$f(n)=o(n!)$