Hamilton cycle and path. Complexity and algorithms.

- Hamiltonian path is a path that visits each vertex exactly once.
- Hamiltonian cycle is a cycle that visits each vertex exactly once and ends at the starting vertex.
- This problem of finding the Hamiltonian cycle is strongly NP-complete, however an algorithm that uses backtracking can be used:

```
def hamiltonian(graph):
    path = [0]
    def dfs(path):
        if len(path) == len(graph) and graph[path[-1]][path[0
            return True, path
        for i in range(len(graph)):
            if i in path: continue
            if graph[path[-1]][i] != 1: continue
            new_path = path.copy()
            new_path.append(i)
            result = dfs(new_path)
            if result[0]: return True, result[1]
        return False, []

    return dfs(path)
```

- This algorithm has the complexity $O(n!)$