

Báo cáo bài thực hành buổi 1

1. Breadth First Search

Mê cung	Số nút đã duyệt	Đường đi (giải pháp)	Có phải là giải pháp tối ưu không?
Nhỏ (tiny)	15	8 bước ['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Có
Trung bình (medium)	269	68 bước ['West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West']	Có
Lớn (big)	620	210 bước ['North', 'North', 'West', 'West', 'West', 'West', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'East', 'East', 'North', 'North', 'East', 'East', 'East', 'East', 'North', 'North', 'East', 'East', 'South', 'South', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'West', 'West', 'North', 'North', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East']	Có

		'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'West', 'West', 'South', 'South', 'South']	
--	--	---	--

2. Depth Fist Search

Mê cung	Số nút đã duyệt	Đường đi (giải pháp)	Có phải là giải pháp tối ưu không?
Nhỏ (tiny)	15	10 bước ['West', 'West', 'West', 'West', 'South', 'South', 'East', 'South', 'South', 'West'.]	Không
Trung bình (medium)	146	130 bước ['West', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'East', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'South', 'South', 'South', 'South', 'East', 'East', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'West', 'West', 'West', 'West', 'West', 'West']	Không
Lớn (big)	390	210 bước	Không

		['North', 'North', 'West', 'West', 'West', 'West', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'East', 'East', 'North', 'North', 'East', 'East', 'East', 'East', 'North', 'North', 'East', 'East', 'South', 'South', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'South', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'South', 'West', 'West', 'North', 'North', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'North', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'South', 'East', 'East', 'South', 'South', 'West', 'West', 'South', 'South', 'South', 'South', 'West', 'West', 'South', 'South']	
--	--	---	--

3. Source code

```

from util import Stack
def depthFirstSearch(problem):

    s = Stack()
    v = set()

    s.push((problem.getStartState(), []))

```

```

while not s.isEmpty():
    state, path = s.pop()

    if problem.isGoalState(state):
        print("buoc: ")
        print(len(path))
        with open("dfs_path.txt", "w") as f:
            for hd in path:
                f.write("'" + hd + "'", " ")
        return path

    if state not in v:
        v.add(state)
        for sc, hd, cost in problem.getSuccessors(state):
            if sc not in v:
                s.push((sc, path + [hd]))

return []

```

```

def breadthFirstSearch(problem):

    f = util.Queue()
    start = problem.getStartState()
    f.push((start, []))

    v = set()

    while not f.isEmpty():
        state, path = f.pop()

        if state in v:
            continue

        v.add(state)

        if problem.isGoalState(state):
            print("buoc: ")
            print(len(path))
            with open("bfs_path.txt", "w") as f:
                for hd in path:
                    f.write("'" + hd + "'", " ")
            return path

        for sc, hd, stepCost in problem.getSuccessors(state):
            if sc not in v:
                f.push((sc, path + [hd]))

    return []

```

