Trần Quang Minh B2304069

Thực hành buổi 1

Bài tập

	Depth First Search			Breath First Search		
Mê cung	Số nút đã duyệt	Đường đi (giải pháp)	Có phải là giải pháp tối ưu không?	Số nút đã duyệt	Đường đi (giải pháp)	Có phải là giải pháp tối ưu không (?)
Tiny	15	['West', 'West', 'West', 'West', 'South', 'South', 'East', 'South', 'South', 'West']	Không phải (Score: 500)	15	['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Phải (Score: 502)
Medium	146	['West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', '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', 'West', 'West', 'West', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West']	Không phải (Score: 380)	269	['West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', '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', 'West', 'West', 'West', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West']	Phải (Score: 442)
Big	390	['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', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'East', 'East',	Phải (Score: 300)	620	['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', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'East', 'East',	Phải (Score: 300)

		'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']			'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']	

Source code

```
def depthFirstSearch(problem):
    from game import Directions
    from util import Stack

    trace = {}
    trace[problem.getStartState()] = (-1, -1)
    marked = {}
    s = Stack()
    s.push(problem.getStartState())

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

        if problem.isGoalState(state):
            path = []
            while (state != problem.getStartState()):
                direct = trace[state][1]
                if direct == 'West':
                    path.append(Directions.WEST)
                elif direct == 'South':
                    path.append(Directions.SOUTH)
                elif direct == 'North':
                    path.append(Directions.NORTH)
                elif direct == 'East':
                    path.append(Directions.EAST)
                state = trace[state][0]

            path.reverse()
            print(path)
            return path

        if state in marked:
            continue
        marked[state] = 1

        for next_state, direct, _ in problem.getSuccessors(state):
```

```
            if next_state in marked:
                continue
            trace[next_state] = [state, direct]
            s.push(next_state)

    return []

def breadthFirstSearch(problem):
    from game import Directions
    from util import Queue

    trace = {}
    trace[problem.getStartState()] = (-1, -1)
    q = Queue()
    q.push(problem.getStartState())

    while not q.isEmpty():
        state = q.pop()

        if problem.isGoalState(state):
            path = []
            while (state != problem.getStartState()):
                direct = trace[state][1]
                if direct == 'West':
                    path.append(Directions.WEST)
                elif direct == 'South':
                    path.append(Directions.SOUTH)
                elif direct == 'North':
                    path.append(Directions.NORTH)
                elif direct == 'East':
                    path.append(Directions.EAST)
                state = trace[state][0]

            path.reverse()
            print(path)
            return path

        for next_state, direct, _ in problem.getSuccessors(state):
```

```
            if next_state in trace:
                continue
            trace[next_state] = [state, direct]
            q.push(next_state)

    return []
```