Trần Quang Minh B2304069

Thực hành buổi 2

Bài tập 1

	Depth First Search			Breath First Search			Uniform Cost 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 ?	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)	15	['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Phải (Score: 502)
Medium	146	['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', 'East', 'South', 'South',	Không phải (Score: 380)	269	['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', 'East', 'South', 'South',	Phải (Score: 442)	269	['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', 'East', 'South', 'South',	Phải (Score: 442)

		'South', 'South', 'South', 'South', 'South', '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', 'West']			'South', 'South', 'South', 'South', 'South', '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', 'West']			'South', 'South', 'South', 'South', 'South', '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', 'West']	
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', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', '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', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', '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', 'West', 'North', 'North', 'East', 'East', 'North', 'North', 'West', 'West', 'North', 'North', 'North', 'North', 'North', 'North', 'East', 'East', 'East', 'East', 'East',	Phải (Score: 300)

🟩		'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',			'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',			'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',			'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',			'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']			'South', 'South', 'South', 'West', 'West', 'South', 'South']			'South', 'South', 'South', 'West', 'West', 'South', 'South']	

Source code

```python
def uniformCostSearch(problem):
    """Search the node of least total cost first."""
    "*** YOUR CODE HERE ***"
    from game import Directions
    from util import PriorityQueue

    trace = {}
    trace[problem.getStartState()] = (-1, -1)
    marked = {}
    pq = PriorityQueue()
    pq.push((problem.getStartState(), 0), 0)
    while not pq.isEmpty():
        state, costSoFar = pq.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, stepCost in problem.getSuccessors(state):
            if next_state in marked:
                continue
            trace[next_state] = [state, direct]
            pq.push((next_state, costSoFar + stepCost), costSoFar + stepCost)


    return []
```

Bài tập 2

	A*			Breath First Search			Uniform Cost 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 ?	Số nút đã duyệt	Đường đi (giải pháp)	Có phải là giải pháp tối ưu không ?
Tiny	15	['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Phải (Score: 502)	15	['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Phải (Score: 502)	15	['South', 'South', 'West', 'South', 'West', 'West', 'South', 'West']	Phải (Score: 502)
Medium	269	['West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South',	Phải (Score: 442)	269	['West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South',	Phải (Score: 442)	269	['West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'South', 'South',	Phải (Score: 442)

		'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'South', 'South', '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', 'West']			'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'South', 'South', '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', 'West']			'East', 'East', 'South', 'South', 'South', 'West', 'West', 'West', 'North', 'West', 'West', 'West', 'West', 'South', 'South', 'South', 'East', 'East', 'East', 'East', 'East', 'East', 'East', 'South', 'South', 'South', 'South', 'South', 'South', 'South', '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', 'West']	
Big	620	['North', 'North', 'West', 'West', 'West', 'West', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West',	Phải (Score: 300)	620	['North', 'North', 'West', 'West', 'West', 'West', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West',	Phải (Score: 300)	620	['North', 'North', 'West', 'West', 'West', 'West', 'North', 'North', 'West', 'West', 'South', 'South', 'West', 'West', 'West', 'West', 'West', 'West', 'West', 'West',	Phải (Score: 300)

🟩		'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', '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', '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',			'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',			'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']			'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']			'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']	

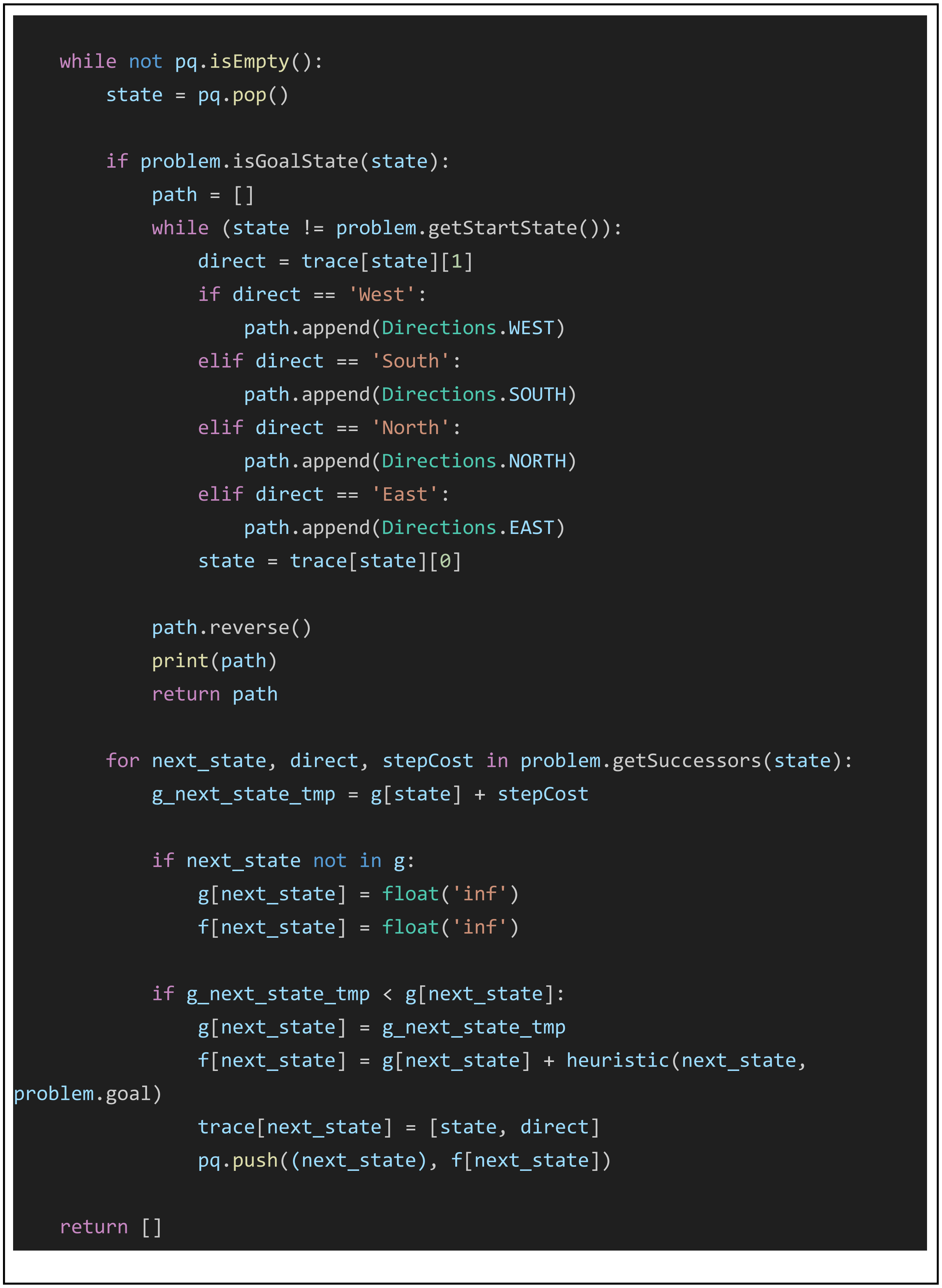
Source code

```
def aStarSearch(problem, heuristic=manhattanHeuristic):
    """Search the node that has the lowest combined cost and heuristic
first."""
    "*** YOUR CODE HERE ***"
    from game import Directions
    from util import PriorityQueue

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

    g = {}
    g[problem.getStartState()] = 0
    f = {}
    f[problem.getStartState()] = heuristic(problem.getStartState(),
problem.goal)

    pq = PriorityQueue()
    pq.push((problem.getStartState()), 0)
```



```
    while not pq.isEmpty():
        state = pq.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, stepCost in problem.getSuccessors(state):
            g_next_state_tmp = g[state] + stepCost

            if next_state not in g:
                g[next_state] = float('inf')
                f[next_state] = float('inf')

            if g_next_state_tmp < g[next_state]:
                g[next_state] = g_next_state_tmp
                f[next_state] = g[next_state] + heuristic(next_state,
problem.goal)
                trace[next_state] = [state, direct]
                pq.push((next_state), f[next_state])

    return []
```