

REPORT CÀI ĐẶT CÁC THUẬT TOÁN BFS, UCS, A*

DEALINE VỀ NHÀ TUẦN 03

Họ tên: Nguyễn Sanh Tuấn.

MSSV: 1760227.

Lớp: 17CK2.

Môn: CTTTMNT&ƯD_ thực hành ca 3 chiều t7.

1. Thuật toán BFS.

- Diễn giải về các bước tìm kiếm của thuật toán này
Trước tiên hãy tìm kiếm các nút nông nhất trong cây tìm kiếm.
Hàm này đẩy các nút không truy cập vào hàng đợi.
Các nút được bật từng cái một và các bước sau đây được thực hiện:
 - 1) Nút được đánh dấu là đã truy cập.
 - 2) Nếu đó là nút mục tiêu, vòng lặp dừng và giải pháp thu được bằng cách quay lại bằng cách sử dụng cha mẹ được lưu trữ.
 - 3) Nếu nó không phải là nút mục tiêu, nó được mở rộng.
 - 4) Nếu nút kế tiếp không được truy cập và chưa được mở rộng khi là con của nút khác, sau đó nó được đẩy lên hàng đợi và cha mẹ của nó được lưu trữ.
- Phần cài đặt code của bản thân.

```
#khởi tạo tham số đầu vào là problem
# lấy trạng thái bắt đầu để kiểm tra có phải là đích đến không
start = problem.getStartState()
exploredState = []
exploredState.append(start)
states = util.Queue()
stateTuple = (start, [])
states.push(stateTuple)
#vòng lặp kiểm tra nếu state có phải là đích đến
while not states.isEmpty():
    state, action = states.pop()
# nếu là đích đến trả về kết quả
    if problem.isGoalState(state):
        return action
#mở trạng thái successor(successor, action, stepCost), đi đến trạng thái state
    successor = problem.getSuccessors(state)
    for i in successor:
        coordinates = i[0]
        if not coordinates in exploredState:
            direction = i[1]
            exploredState.append(coordinates)
            states.push((coordinates, action + [direction]))
    return action

util.raiseNotDefined()
```

- Chạy kiểm tra kết quả cài đặt thuật toán.

Chạy các lệnh:

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
```

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=bfs
```

- Kết quả (màn hình)

```
Windows PowerShell
PS E:\CTTTMNTA\UD 2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l mediumMaze -p SearchAgent -a fn=bfs
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores: 442.0
Win Rate: 1/1 (1.00)
Record: Win
PS E:\CTTTMNTA\UD 2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=bfs
[SearchAgent] using function bfs
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 620
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores: 300.0
Win Rate: 1/1 (1.00)
Record: Win
```

2. Thuật toán UCS

- Diễn giải thuật toán
Tìm kiếm nút có tổng chi phí tối thiểu trước.
Hàm này đẩy các nút không truy cập vào hàng ưu tiên.
Các nút được bật từng cái một và các bước sau đây được thực hiện:
1) Nút được đánh dấu là đã truy cập.
2) Nếu đó là nút mục tiêu, vòng lặp dừng và giải pháp thu được bằng cách quay lại bằng cách sử dụng cha mẹ được lưu trữ.
3) Nếu nó không phải là nút mục tiêu, nó được mở rộng.
4) Nếu nút kế tiếp không được truy cập, chi phí của nó được tính.
5) Nếu chi phí của nút kế tiếp được tính toán trước đó trong khi mở rộng một nút khác, và nếu chi phí tính toán mới thấp hơn chi phí cũ, thì chi phí và phụ huynh được cập nhật, và nó được đẩy lên hàng đợi ưu tiên với chi phí mới là ưu tiên.
• Phần code

Các phần tử trong PriorityQueue sẽ là bộ ba (state, path, cost). Trong đó state, path giống như bài DFS. cost sẽ lưu chi phí từ trạng thái đầu đến trạng thái hiện tại.

```
#khởi tạo trạng thái bắt đầu
start = problem.getStartState()
exploredState = []
states = util.PriorityQueue()
states.push((start, []), 0)
#duyệt vòng lặp nếu state chưa phải là đích đến
while not states.isEmpty():
    state, actions = states.pop()
    if problem.isGoalState(state):
        return actions
    if state not in exploredState:
        successors = problem.getSuccessors(state)
        for succ in successors:
            coordinates = succ[0]
            if coordinates not in exploredState:
                directions = succ[1]
```

```

        newCost = actions + [directions]
        states.push((coordinates, actions + [directions]), problem
.getCostOfActions(newCost))
        exploredState.append(state)
    return actions

util.raiseNotDefined()

```

- Kiểm tra kết quả cài đặt thuật toán.

Chạy các lệnh:

```
python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
```

```
python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
```

```
python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
```

- Kết quả (màn hình):

```

PS E:\CTTTMNT4UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l mediumMaze -p SearchAgent -a fn=ucs
(SearchAgent) using function ucs
(SearchAgent) using problem type PositionSearchProblem
Path found with total cost of 68 in 0.0 seconds
Search nodes expanded: 269
Pacman emerges victorious! Score: 442
Average Score: 442.0
Scores: 442.0
Win Rate: 1/1 (1.00)
Record: Win
PS E:\CTTTMNT4UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l mediumDottedMaze -p StayEastSearchAgent
Path found with Total cost of 1 in 0.0 seconds
Search nodes expanded: 186
Pacman emerges victorious! Score: 646
Average Score: 646.0
Scores: 646.0
Win Rate: 1/1 (1.00)
Record: Win
PS E:\CTTTMNT4UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l mediumScaryMaze -p StayWestSearchAgent
Path found with Total cost of 68719479864 in 0.0 seconds
Search nodes expanded: 108
Pacman emerges victorious! Score: 418
Average Score: 418.0
Scores: 418.0
Win Rate: 1/1 (1.00)
Record: Win
PS E:\CTTTMNT4UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3>

```

3. Thuật toán A*

- Diễn giải thuật toán.

Tìm kiếm nút có chi phí kết hợp thấp nhất và heuristic trước.

Hàm này đẩy các nút không truy cập vào hàng ưu tiên.

Các nút được bật từng cái một và các bước sau đây được thực hiện:

- 1) Nút được đánh dấu là đã truy cập.
- 2) Nếu đó là nút mục tiêu, vòng lặp dừng và giải pháp thu được bằng cách quay lại bằng cách sử dụng cha mẹ được lưu trữ.
- 3) Nếu nó không phải là nút mục tiêu, nó được mở rộng.
- 4) Nếu nút kế tiếp không được truy cập, chi phí của nó được tính bằng hàm heuristic.
- 5) Nếu chi phí của nút kế tiếp được tính toán trước đó trong khi mở rộng một nút khác, và nếu chi phí tính toán mới thấp hơn chi phí cũ, thì chi phí và cha mẹ được cập nhật, và nó được đẩy lên hàng đợi ưu tiên với chi phí mới là ưu tiên.

- Phần code .

```

#khởi tạo trạng thái bắt đầu
start = problem.getStartState()
exploredState = []
states = util.PriorityQueue()
states.push((start, []), nullHeuristic(start, problem))
#gán chi phí bằng 0
nCost = 0
#vòng lặp chưa tìm được đích đến
while not states.isEmpty():
    state, actions = states.pop()
    if problem.isGoalState(state):

```

```

        return actions
    if state not in exploredState:
        successors = problem.getSuccessors(state)
        for succ in successors:
            coordinates = succ[0]
            if coordinates not in exploredState:
                directions = succ[1]
                nActions = actions + [directions]
                nCost = problem.getCostOfActions(nActions) + heuristic(coordinates, problem)
                states.push((coordinates, actions + [directions]), nCost)
            exploredState.append(state)
    return actions

util.raiseNotDefined()

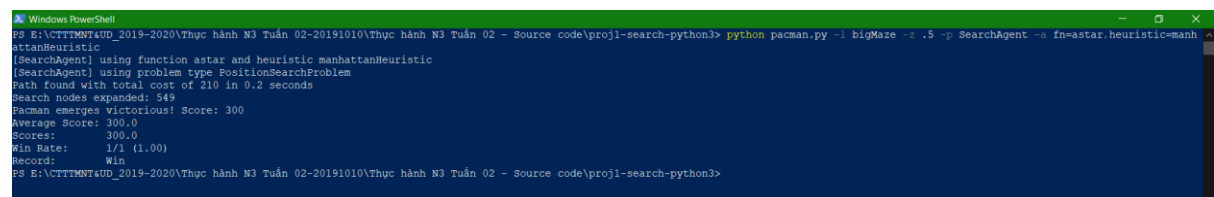
```

- Chạy kiểm tra kết quả cài đặt thuật toán.

Chạy câu lệnh:

```
python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
```

Kết quả màn hình:



```

Windows PowerShell
PS E:\CITTTMNT\UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3> python pacman.py -l bigMaze -z .5 -p SearchAgent -a fn=astar,heuristic=manhattanHeuristic
[SearchAgent] using function astar and heuristic manhattanHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.2 seconds
Search nodes expanded: 549
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores: 300.0
Win Rate: 1/1 (1.00)
Record: Win
PS E:\CITTTMNT\UD_2019-2020\Thực hành N3 Tuần 02-20191010\Thực hành N3 Tuần 02 - Source code\proj1-search-python3>

```

-----THE END-----