

BÁO CÁO KẾT QUẢ THUẬT TOÁN

1. Heuristic cho bài toán ăn hết thức ăn

Trong phần báo toán này, mình sẽ thiết kế 1 class **FoodSearchProblem** trong file *searchAgent.py* để thể hiện bài toán tìm kiếm sau: tìm đường đi ngắn nhất cho Pacman trong mê cung sao cho Pacman đi qua tìm hết thức ăn.

-Hình ảnh minh hoạ cài Class mà cần làm:

```
class FoodSearchProblem:
    """
    food: str problem associated with finding the a path that collects all of the
    food (dots) in a Pacman game.
    A search state in this problem is a tuple ( pacmanPosition, foodGrid ) where
    pacmanPosition: a tuple (x,y) of integers specifying Pacman's position
    foodGrid:      a Grid (see game.py) of either True or False, specifying remaining food
    """

    def __init__(self, startingGameState):
        self.start = (startingGameState.getPacmanPosition(),
                       startingGameState.getFood())
        self.walls = startingGameState.getWalls()
        self.startingGameState = startingGameState
        self._expanded = 0 # DO NOT CHANGE
        self.heuristicInfo = {} # A dictionary for the heuristic to store information

    def getStartState(self):
        return self.start

    def isGoalState(self, state):
        return state[1].count() == 0

    def getSuccessors(self, state):
        "Returns successor states, the actions they require, and a cost of 1."
        successors = []
        self._expanded += 1 # DO NOT CHANGE
        for direction in [Directions.NORTH, Directions.SOUTH, Directions.EAST, Directions.WEST]:
```

```

def getStartState(self):
    return self.start

def isGoalState(self, state):
    return state[1].count() == 0

def getSuccessors(self, state):
    """Returns successor states, the actions they require, and a cost of 1."""
    successors = []
    self._expanded += 1 # DO NOT CHANGE
    for direction in [Directions.NORTH, Directions.SOUTH, Directions.EAST, Directions.WEST]:
        x, y = state[0]
        dx, dy = Actions.directionToVector(direction)
        nextx, nexty = int(x + dx), int(y + dy)
        if not self.walls[nextx][nexty]:
            nextFood = state[1].copy()
            nextFood[nextx][nexty] = False
            successors.append(((nextx, nexty), nextFood), direction, 1))
    return successors

def getCostOfActions(self, actions):
    """Returns the cost of a particular sequence of actions. If those actions
    include an illegal move, return 999999"""
    x, y = self.getStartState()[0]
    cost = 0
    for action in actions:
        # tìm ra trạng thái tiếp theo và xem liệu nó khả thi hay không

```

```

def getCostOfActions(self, actions):
    """Returns the cost of a particular sequence of actions. If those actions
    include an illegal move, return 999999"""
    x, y = self.getStartState()[0]
    cost = 0
    for action in actions:
        # tìm ra trạng thái tiếp theo và xem liệu nó khả thi hay không
        dx, dy = Actions.directionToVector(action)
        x, y = int(x + dx), int(y + dy)
        if self.walls[x][y]:
            return 999999
        cost += 1
    return cost

```

```

#second try:
gs = problem.startingGameState
foodList = foodGrid.asList()
foodCount = len(foodList)
max_dis = 0
part_max_dis = 0

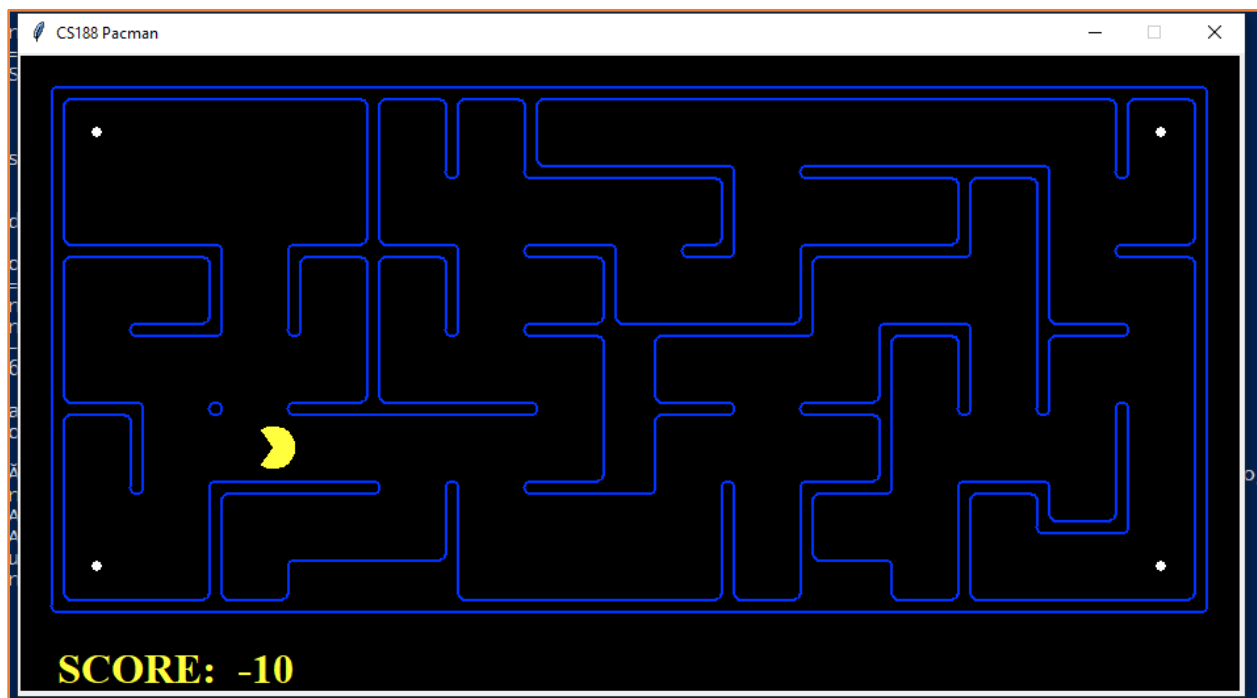
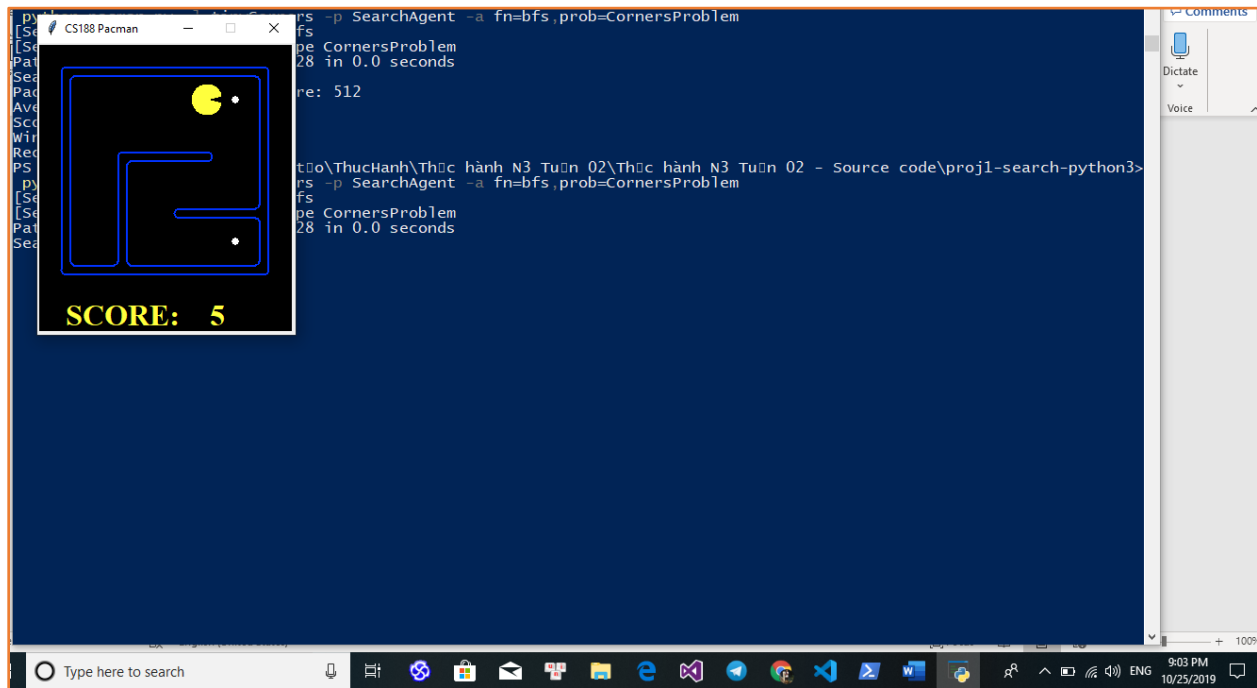
for i in range(foodCount):
    for ii in range(foodCount-i-1):
        dis = mazeDistance(foodList[i], foodList[ii+1], gs)
        if dis > max_dis:
            max_dis = dis
            furthest1 = foodList[i]
            furthest2 = foodList[ii+1]
            part1 = mazeDistance(position, foodList[i], gs)
            part2 = mazeDistance(position, foodList[ii+1], gs)
            if part1 > part2:
                part_max_dis = part2
            else:
                part_max_dis = part1

return max_dis+part_max_dis

```

-Sau khi xây dựng code xong, mình sẽ chạy thực thi đoạn code qua lệnh:

python pacman.py -l trickySearch -p AStarFoodSearchAgent với ***AStarFoodSearchAgent*** là "shortcut" của: ***-p SearchAgent -a "fn=astar,prob=FoodSearchProblem,heuristic=foodHeuristic"***



-Và bạn có thể bạn gõ ***python autograder.py -q q7*** để kiểm tra phần cài đặt của bạn với các bộ test khác nhau.

```

Question q2
=====
*** PASS: test_cases\q2\graph_backtrack.test
***   solution:      ['1:A->C', '0:C->G']
***   expanded_states: ['A', 'B', 'C', 'D']
*** PASS: test_cases\q2\graph_bfs_vs_dfs.test
***   solution:      ['1:A->G']
***   expanded_states: ['A', 'B']
*** PASS: test_cases\q2\graph_infinite.test
***   solution:      ['0:A->B', '1:B->C', '1:C->G']
***   expanded_states: ['A', 'B', 'C']
*** PASS: test_cases\q2\graph_manypaths.test
***   solution:      ['1:A->C', '0:C->D', '1:D->F', '0:F->G']
***   expanded_states: ['A', 'B1', 'C', 'B2', 'D', 'E1', 'F', 'E2']
*** PASS: test_cases\q2\pacman_1.test
***   pacman layout: mediumMaze
***   solution length: 68
***   nodes expanded: 269

### Question q2: 3/3 ###

Question q5
=====
*** PASS: test_cases\q5\corner_tiny_corner.test
***   pacman layout: tinyCorner
***   solution length: 28

### Question q5: 3/3 ###

Finished at 21:05:06

Provisional grades
=====
Question q2: 3/3
Question q5: 3/3
-----
Total: 6/6

```

2. Tìm kiếm nhanh

Sau khi hoàn thành câu 2.7 thì mình tiếp tục đến với thuật toán tìm kiếm nhanh bằng **Heuristic**. Mình sẽ viết hoàn chỉnh hàm **cornersHeuristic** trong file [searchAgent.py](#). Hàm này nhận hai tham số đầu vào là trạng thái state và bài toán problem, và trả về chi phí ước lượng từ state đến đích.

-Hình ảnh minh họa code và note lại những ghi chú trong khi code:

```

def findPathToClosestDot(self, gameState):
    """
    Returns a path (a list of actions) to the closest dot, starting from
    gameState.
    """
    # Here are some useful elements of the startState
    startPosition = gameState.getPacmanPosition()
    food = gameState.getFood()
    walls = gameState.getWalls()
    problem = AnyFoodSearchProblem(gameState)

    """*** You don't need to implement it ***"""
    util.raiseNotDefined()

```

-Sau khi đã cài đặt xong, bạn có thể kiểm tra một cách trực quan trên game Pacman bằng cách gõ câu lệnh:

python pacman.py -l bigSearch -p ClosestDotSearchAgent -z .5

```

PS E:\NAM 3- HK1\Tri tu nhân t\o\ThucHanh\Thưc hành N3 Tuần 02\ii\proj1-search-python3> python pacman.py -l mediumCorners
rs -p AStarCornersAgent -z 0.5

```

-Sau khi chạy lệnh , bạn có thể gõ ***python autograder.py -q q8*** để kiểm tra phần cài đặt của bạn với các bộ test khác nhau:

--→ Kết quả này bị Fail rồi, mình sẽ cố gắng sửa lại thuật toán sớm nhất có thể !

THANK YOU