

2022-HSEA-第一次作业

1. 任务一

任务叙述

在 `search.py` 中的 `aStarSearch` 中实现 A*算法的图搜索函数代码和启发式函数代码，以完成糖豆人到唯一糖豆的路径寻找，即完成 `bigMaze`、`openMaze`、`smallMaze` 三个地图的路径寻找。

解决方法

实现图搜索代码 `aStarSearch`，分别利用曼哈顿距离和欧式距离实现启发式函数设计。实际运行发现，曼哈顿距离的效果要好于欧式距离，这是因为欧式距离计算的是直线距离，而曼哈顿距离是折线距离，本问题中，曼哈顿距离大于等于欧式距离，因此前者的启发式方法优于后者。

```
def aStarSearch(problem, heuristic=nullHeuristic):
    explored = []
    que = util.PriorityQueue()
    que.push([problem.getStartState(), 0, []], 0)

    while que.isEmpty() == False:
        [state, cost, actions] = que.pop()
        if problem.isGoalState(state):
            return actions
        if state not in explored:
            explored.append(state)
            for _state, _action, _cost in problem.getSuccessors(state):
                new_cost = cost + _cost
                new_actions = actions + [_action]
                que.push([_state, new_cost, new_actions], new_cost +
                    heuristic(_state, problem))
    return []
```

曼哈顿距离

```
def myHeuristic(state, problem=None):
    x, y = state
    gx, gy = problem.goal
```

```
return abs(x - gx) + abs(y - gy)
```

欧式距离

```
def myHeuristic(state, problem=None):
    x, y = state
    gx, gy = problem.goal
    return sqrt((x - gx) ** 2 + (y - gy) ** 2)
```

实验效果

任务一中三个地图糖豆人均能找到最优路径，具体扩展结点数和运行截图如下：

曼哈顿距离

```
(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l openMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.0 seconds
Search nodes expanded: 535
Pacman emerges victorious! Score: 456
Average Score: 456.0
Scores: 456.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l smallMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 19 in 0.0 seconds
Search nodes expanded: 53
Pacman emerges victorious! Score: 491
Average Score: 491.0
Scores: 491.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l bigMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 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
```

欧式距离

```
(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l bigMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 210 in 0.0 seconds
Search nodes expanded: 557
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores: 300.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l smallMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 19 in 0.0 seconds
Search nodes expanded: 56
Pacman emerges victorious! Score: 491
Average Score: 491.0
Scores: 491.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l openMaze -p SearchAgent -a fn=astar,heuristic=myHeuristic
[SearchAgent] using function astar and heuristic myHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.0 seconds
Search nodes expanded: 550
Pacman emerges victorious! Score: 456
Average Score: 456.0
Scores: 456.0
Win Rate: 1/1 (1.00)
Record: Win
```

null

```
(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l openMaze -p SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 54 in 0.0 seconds
Search nodes expanded: 682
Pacman emerges victorious! Score: 456
Average Score: 456.0
Scores: 456.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l smallMaze -p SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[SearchAgent] using problem type PositionSearchProblem
Path found with total cost of 19 in 0.0 seconds
Search nodes expanded: 92
Pacman emerges victorious! Score: 491
Average Score: 491.0
Scores: 491.0
Win Rate: 1/1 (1.00)
Record: Win

(base) E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l bigMaze -p SearchAgent -a fn=astar,heuristic=nullHeuristic
[SearchAgent] using function astar and heuristic nullHeuristic
[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
```

结果展示

地图	启发式函数	扩展结点数	得分	耗时
bigMaze	曼哈顿距离	549	300	0
	欧式距离	557	300	0
	null	620	300	0
openMaze	曼哈顿距离	535	456	0
	欧式距离	550	456	0
	null	682	456	0
smallMaze	曼哈顿距离	53	491	0
	欧式距离	56	491	0
	null	92	491	0

必要分析

由于采用图搜索并且两种启发式函数均有 consistent 性质，因此最优性可以得到保证。因而，三个地图均能找到最优路径，但曼哈顿距离启发函数扩展的结点数要小于欧式距离启发函数扩展的结点数，其时间复杂度和空间复杂度均优于后者，这是因为欧式距离计算的是直线距离，而曼哈顿距离是折线距离，本问题中，曼哈顿距离大于等于欧式距离，因此前者的启发式方法优于后者。

并且两种方式均优于 null 启发式函数，这是因为两种距离均大于等于零，因而优于 null 启发式函数，并且对于越简单的地图，扩展结点减少的幅度越大，对于复杂地图，曼哈顿和欧氏距离启发式函数的效果有限。

2 任务二

任务叙述

任务二为多结点的搜索问题，在尽可能少的步骤中吃掉所有的豆子，引入新的搜索问题定义来形式化食物清理问题，即 `searchAgents.py` 中的 `FoodSearchProblem`。需要完成 `searchAgents.py` 中 `foodHeuristic` 函数的编写，找到一条收集 Pacman 世界中所有食物的路径，进而帮助吃豆人进行游戏。

解决方法

实现启发式函数代码 `foodHeuristic`，问题 2 和问题 1 的不同之处在于食物数量由一个变为多个，因此启发式函数最好可以利用到食物数量的信息，简单起见，我们分两种情况进行讨论，首先是特例，即食物已经被吃完，此时启发式函数值应为零；然后是普通情况，这时我们计算当前位置和最近食物的曼哈顿距离，然后加上剩余食物的数量减一，此启发式函数的 `admissible` 和 `consistent` 性质证明见必要分析部分。

$$h = dis_{min} + num_{food} - 1$$

```
# 可以证明启发式的算法
def foodHeuristic(state, problem):
    position, foodGrid = state
    foodList = foodGrid.asList()
    if len(foodList) == 0:
        return 0
    curx, cury = position
    distances = []
    for x, y in foodList:
        distances.append(abs(x - curx) + abs(y - cury))
    closest = min(distances)
    return closest + len(foodList) - 1
```

实验效果

任务二中三个地图糖豆人的具体扩展结点数和运行截图如下，其中启发式函数为零时，对于地图三在十分钟后仍无法计算出结果。

启发式函数

```
E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search1 -p AStarFoodSearchAgent
Path found with total cost of 6 in 0.0 seconds
Search nodes expanded: 7
Pacman emerges victorious! Score: 534
Average Score: 534.0
Scores: 534.0
Win Rate: 1/1 (1.00)
Record: Win

E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search2 -p AStarFoodSearchAgent
Path found with total cost of 16 in 0.0 seconds
Search nodes expanded: 72
Pacman emerges victorious! Score: 614
Average Score: 614.0
Scores: 614.0
Win Rate: 1/1 (1.00)
Record: Win

E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search3 -p AStarFoodSearchAgent
Path found with total cost of 31 in 0.0 seconds
Search nodes expanded: 31
Pacman emerges victorious! Score: 779
Average Score: 779.0
Scores: 779.0
Win Rate: 1/1 (1.00)
Record: Win
```

null

```
E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search1 -p AStarFoodSearchAgent
Path found with total cost of 6 in 0.0 seconds
Search nodes expanded: 14
Pacman emerges victorious! Score: 534
Average Score: 534.0
Scores: 534.0
Win Rate: 1/1 (1.00)
Record: Win

E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search2 -p AStarFoodSearchAgent
Path found with total cost of 16 in 0.1 seconds
Search nodes expanded: 707
Pacman emerges victorious! Score: 614
Average Score: 614.0
Scores: 614.0
Win Rate: 1/1 (1.00)
Record: Win

E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l Search3 -p AStarFoodSearchAgent
```

结果展示

地图	启发式函数	扩展结点数	得分	耗时	节点数减少值
search1	启发式函数	7	534	0s	7
	null	14	534	0s	0
search2	启发式函数	72	614	0s	635
	null	707	614	0.1s	0
search3	启发式函数	31	779	0s	?
	null	?	?	>10min	?

由上表对比可知，设计的启发式算法大大减少了搜索所扩展的节点数，尤其是对于最复杂的地图三；并且地图越复杂，搜索节点减少的幅度越大。

必要分析

关于所设计启发式函数的 admissible 和 consistent 性质证明过程如下：

admissible 性质需要满足 $0 \leq h(n) \leq h^*(n)$ ，我们设计的启发式函数为：没有食物时值为零，有食物时为当前位置和最近食物的曼哈顿距离加上剩余食物的数量减一；易知两者都

大于等于零，而对于不等式右边，只有当所有食物都连在一起并且糖豆人可以不绕路到达最近的食物点时等式才成立，其余情况启发式函数值均小于最优路径的真实代价值。

consistent 性质需要满足 $h(n) \leq c(n, a, n') + h(n')$ ，假设当前的食物数量为 L ，我们分两种情况讨论，第一种情况是下一步会吃到食物，那么这一步的 $h(n) = 1 + L - 1 = L$ ，（因为只有曼哈顿距离为 1 时才能下一步吃到食物）， $c(n, a, n') = 1$ ，下一步的 $h(n') = dis + (L - 1) - 1 = dis + L - 2$ ，那么 $c(n, a, n') + h(n') = dis + L - 1$ ，由于 $dis \geq 1$ （ dis 为和最近食物的曼哈顿距离），所以 $c(n, a, n') + h(n') \geq L = h(n)$ ，即满足性质；第二种情况是下一步不会吃到食物， $c(n, a, n') = 1$ ，由于食物数量不变，因此如果下一步距离最近的食物更近一步，那么 $h(n) - h(n') = 1$ ，其余情况均为 $h(n) - h(n') < 1$ ，因而 $h(n) \leq 1 + h(n') = c(n, a, n') + h(n')$ ，此时也满足性质。

综上所述，所设计的启发式函数满足 admissible 和 consistent 性质。

3 任务三

任务叙述

任务二中的解决方案只取决于墙壁、普通食物和吃豆人的位置，没有考虑幽灵和能量食物的作用。任务三需要在 minimaxClassic、originalClassic、powerClassic 三个有幽灵的布局上运行任务二中的 A*搜索算法。

解决方法

将任务二的解决方法直接应用于任务三的布局。

实验效果与必要分析

任务三中糖豆人在 minimaxClassic 地图的具体扩展结点数和运行截图如下：扩展节点数为 6、得分为 516、运行时间为 0s。

```
E:\master\研一课程\5 启发式搜索与演化算法\作业\HESA_assignment1\search-code>python pacman.py -l minimaxClassic -p AStarFoodSearchAgent -n 5
Path found with total cost of 4 in 0.0 seconds
Search nodes expanded: 6
Pacman emerges victorious! Score: 516
Path found with total cost of 4 in 0.0 seconds
Search nodes expanded: 6
Pacman emerges victorious! Score: 516
Path found with total cost of 4 in 0.0 seconds
Search nodes expanded: 6
Pacman emerges victorious! Score: 516
Path found with total cost of 4 in 0.0 seconds
Search nodes expanded: 6
Pacman emerges victorious! Score: 516
Path found with total cost of 4 in 0.0 seconds
Search nodes expanded: 6
Pacman emerges victorious! Score: 516
Average Score: 516.0
Scores:      516.0, 516.0, 516.0, 516.0, 516.0
Win Rate:    5/5 (1.00)
Record:      Win, Win, Win, Win, Win
```

糖豆人在 `originalClassic` 和 `powerClassic` 地图运行后无响应，超过十分钟仍未出结果，这可能是因为食物数量过多且地图过于复杂的缘故，并且由于幽灵和能量食物的存在，如果幽灵是随机游走的，那么启发式算法的最优性也无法得到保障。

4 操作说明

依照发布的作业说明文档输入命令运行即可；

对于任务一，如果想使用空启发式函数不指定 `heuristic` 参数即可，代码默认使用曼哈顿距离，若想使用欧氏距离需要将 `myHeuristic` 函数中的欧式距离代码行取消注释即可；

对于任务二，如果想使用空启发式函数直接将 `foodHeuristic` 函数中的 `return 0` 语句取消注释即可。