

# Assignment 1

- capture the result of `autograder.py` in terminal

```
Finished at 10:31:13

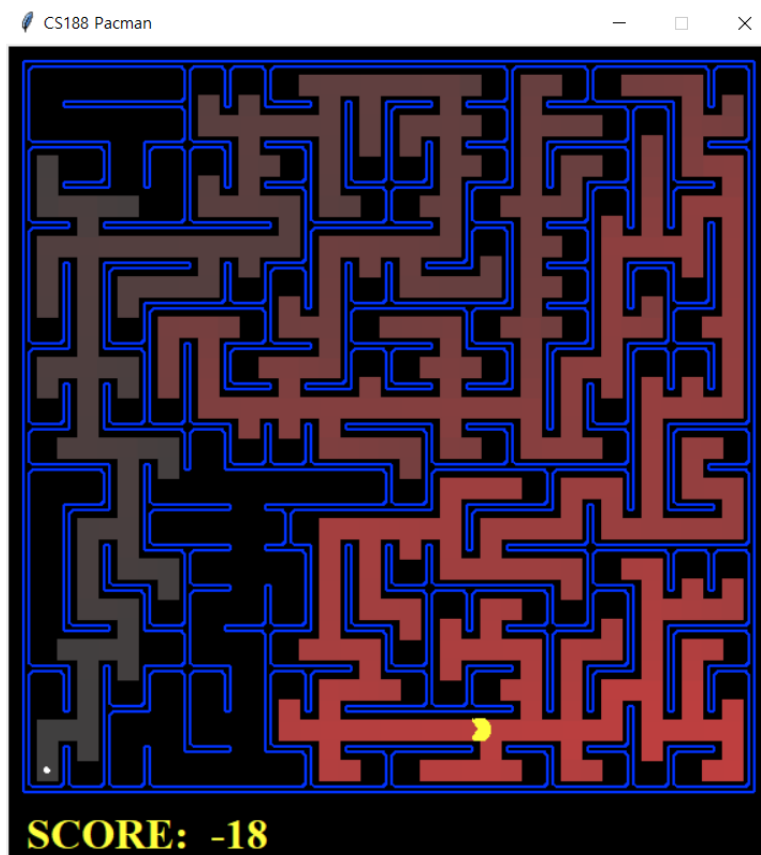
Provisional grades
=====
Question q1: 3/3
Question q2: 3/3
Question q3: 3/3
Question q4: 3/3
Question q5: 0/3
Question q6: 0/3
Question q7: 0/4
Question q8: 0/3
-----
Total: 12/25
```

- Discuss which algorithm is better between DFS or BFS. (In mediumMaze)
  - BFS is better. DFS takes less time than BFS (node expanded 146 v.s. 269). When it comes to optimumness of the solution, however, DFS is terrible (score 380 v.s. 442). Therefore, BFS is better in mediumMaze case.
- The proposed heuristic function is Manhattan. Is there any other heuristic function that is more efficient? Discuss specific cases where your algorithm works effectively.
  - If we can know the number of successors, we can remove node that has 1 successor if that successor is not solution. However, if we should expand the node to know the number of successors, it is not efficient.
    - In pacman case, we can know the number of successors using `problem.walls` without expanding nodes. Practically, however, it counts the number of successors node by node, therefore there is overhead. But still, in the problem where we can know the number of successors, we can use this technique.
  - Additionally, we can improve heuristic when we can know whether or not there is a wall at a specific location. If there is wall between current position and goal, then optimal distance increase by 2. See image below. In that case, distance must be  $length\ of\ maze + 2$ . So if there is wall at the direction of pacman to dot, we can improve distance function slightly.



3. Is my heuristic is admissible? If not, is it okay?

- It is not admissible since it returns over `9999999999999999` value when the number of successors is one and it is not a solution. However, it still guarantees finding an optimal solution. Because if the number of successors is one and it is not a solution, it guarantees that the non-admissible case is not the solution.
- Capture the result of A\* algorithm with your implemented heuristic function.



```
$ python pacman.py -l bigMaze -z .5 -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: 468
Pacman emerges victorious! Score: 300
Average Score: 300.0
Scores:      300.0
Win Rate:    1/1 (1.00)
Record:      win
(2023AI)
```