# Week 4:

Each search algorithm uses a different data structure from the util.py file

DFS uses Stack, BFS uses queue while UCS and A\* use Priority Queues

The UCS and A\* algorithms follow similar approaches, but A\* also accounts for heuristic values associated with nodes.

In UCS the basic idea is that we keep track of the costs to traverse to the successors of unvisited nodes and store this in queues (priority queues in case of UCS and A\*).

The priority queue considers the costs to assign priority, with lower costs being given higher priority.

**Algorithm**

The algorithms store the start node information as the current node in "leaf". In UCS and A\* priority/ cost 0 is assigned to it.

Till the goal node is not reached, the successors are kept track of.

In the case of UCS and A\* this is pushed to a priority list and is internally sorted according to descending priority order and it is checked whether these are previously visited.

Then the current node is updated to reflect the current newly visited node and the cost to traverse to these nodes is calculated.

The corresponding data structure elements are then popped at the end.

The BFS and DFS algorithm differ majorly in that one uses Queue (FIFO logic) while the other uses Stack (FILO logic)

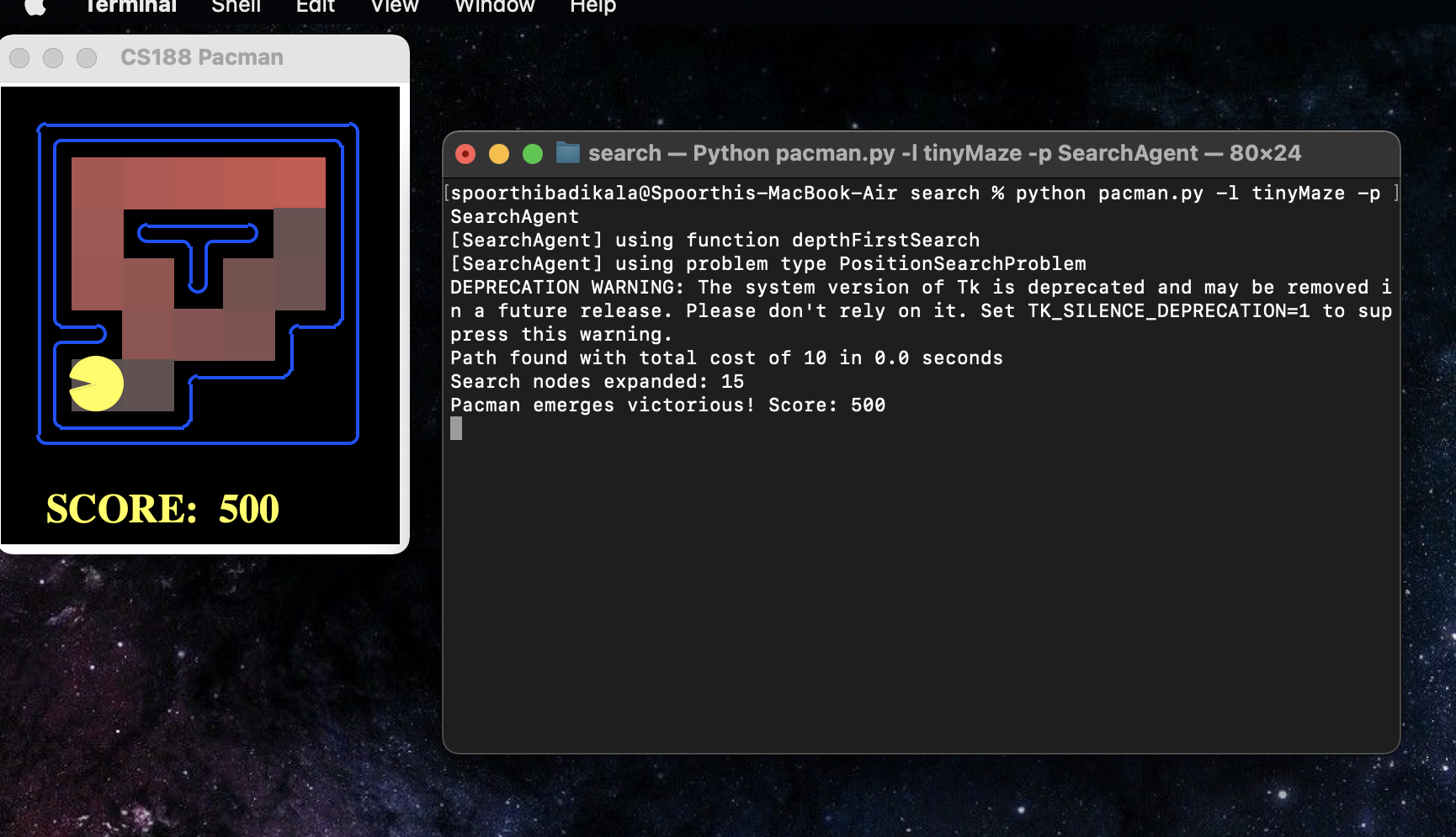
Our key learnings included implementing the complete search algorithms for a real time application

**Question 1:**

The DFS algorithm does not offer the optimum solution. Due to the nature of the algorithm, it explores each branch all the way down to the leaf nodes. This sometimes causes unnecessary exploration in paths that will not lead to the goal (for example, search down a branch that does not lead to goal)

A\* is optimal since it is an informed search and has heuristic values corresponding to each node

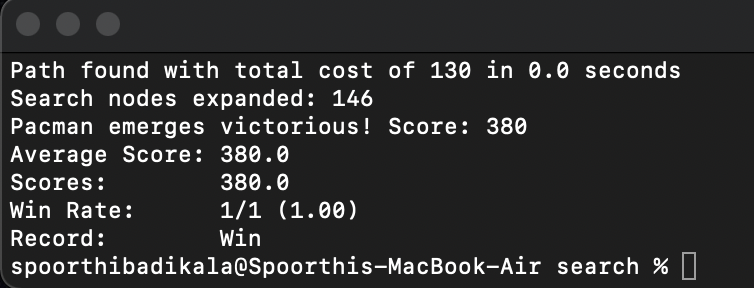
# Screenshots

Q1. A. 

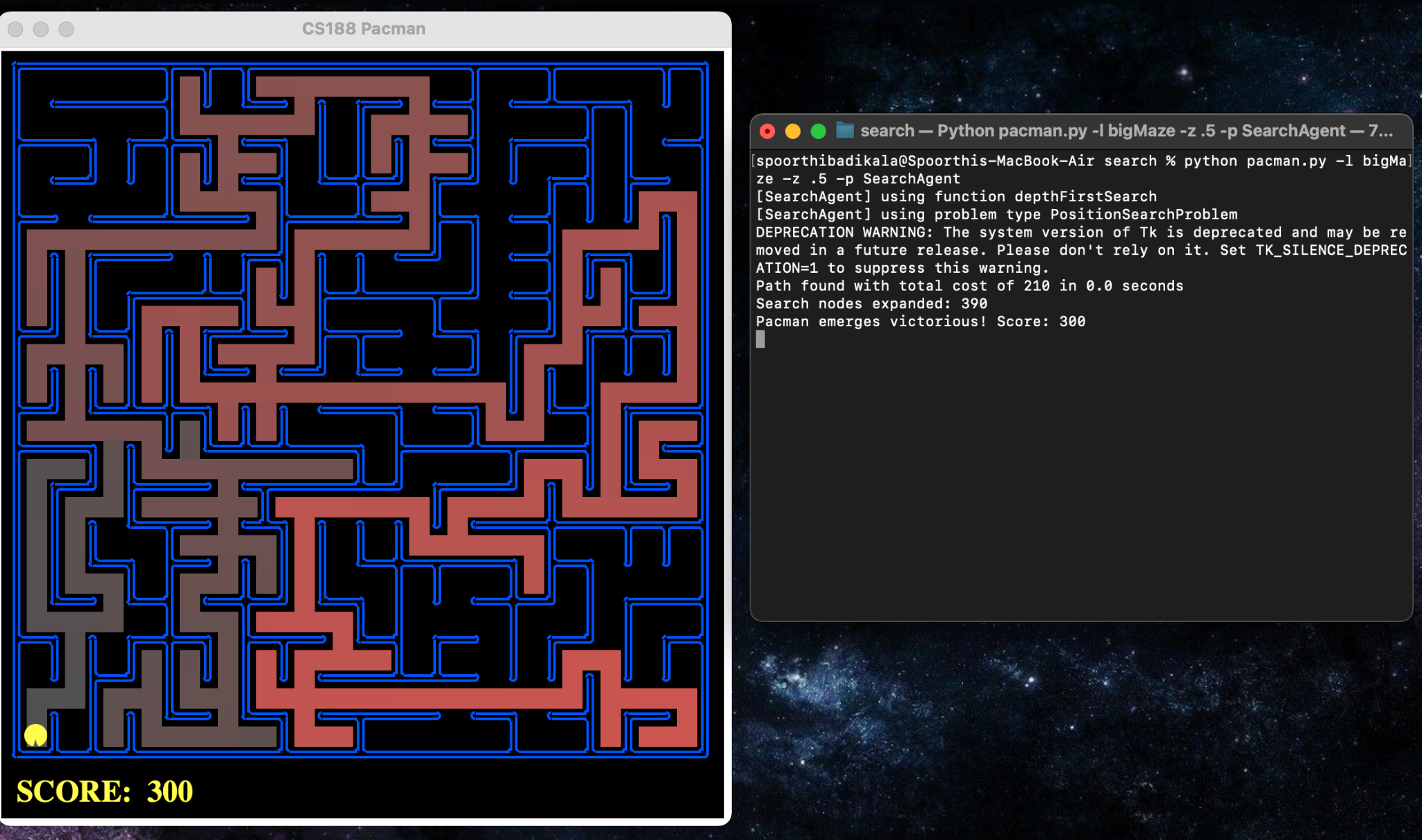


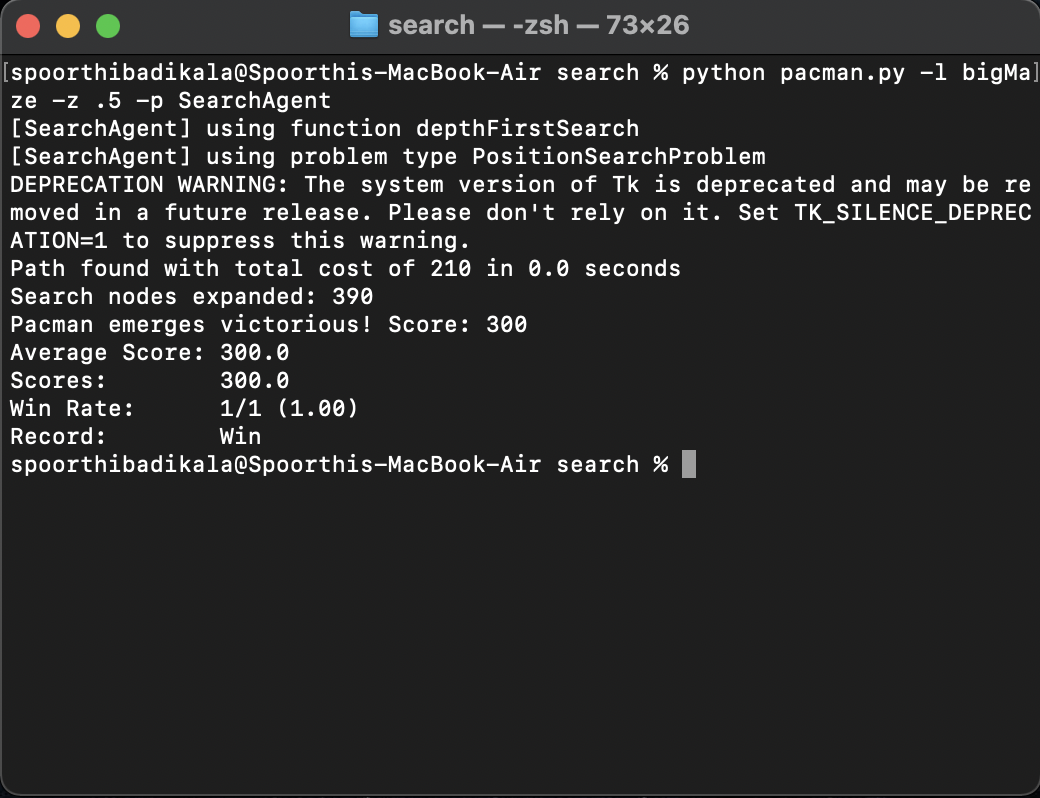
Q1. B.



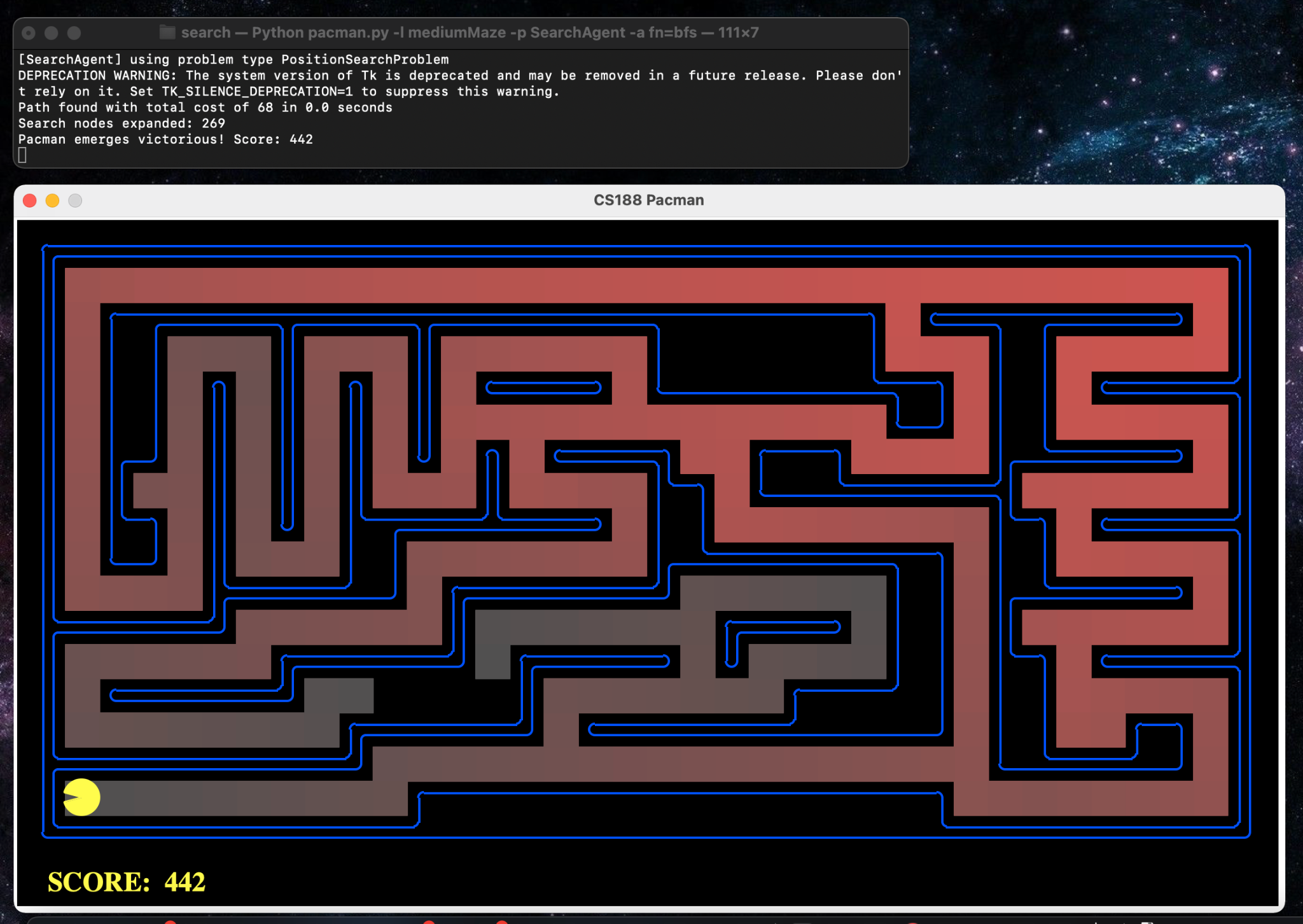


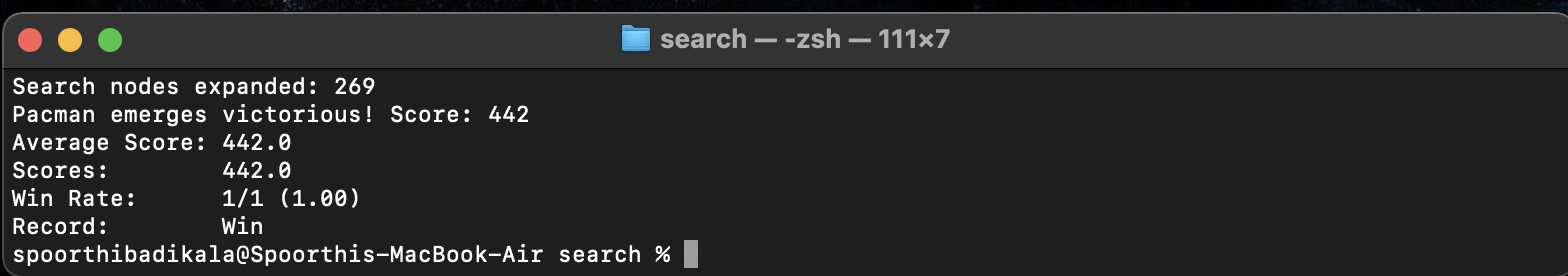
Q1. C.



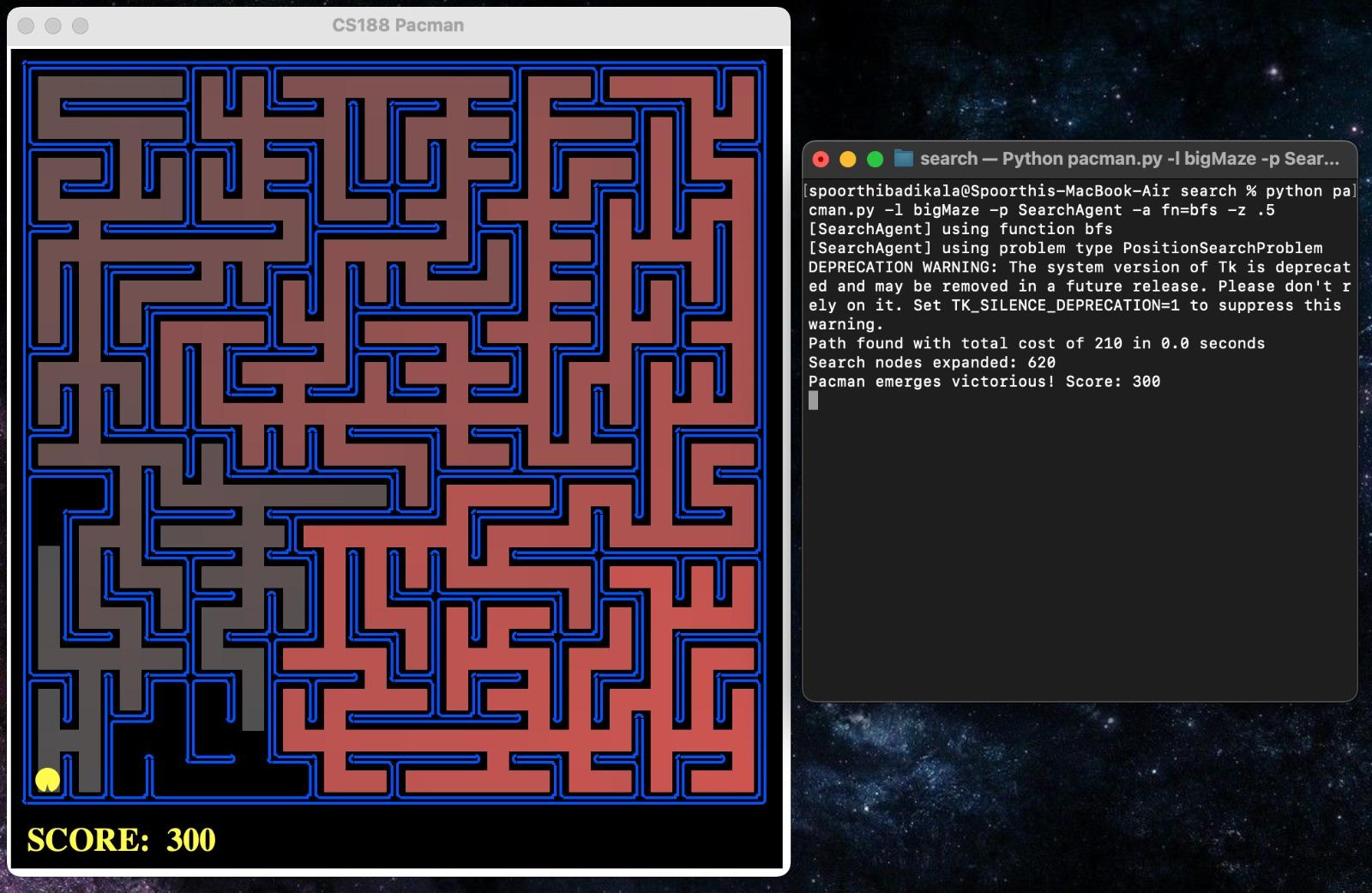


Q2. A.



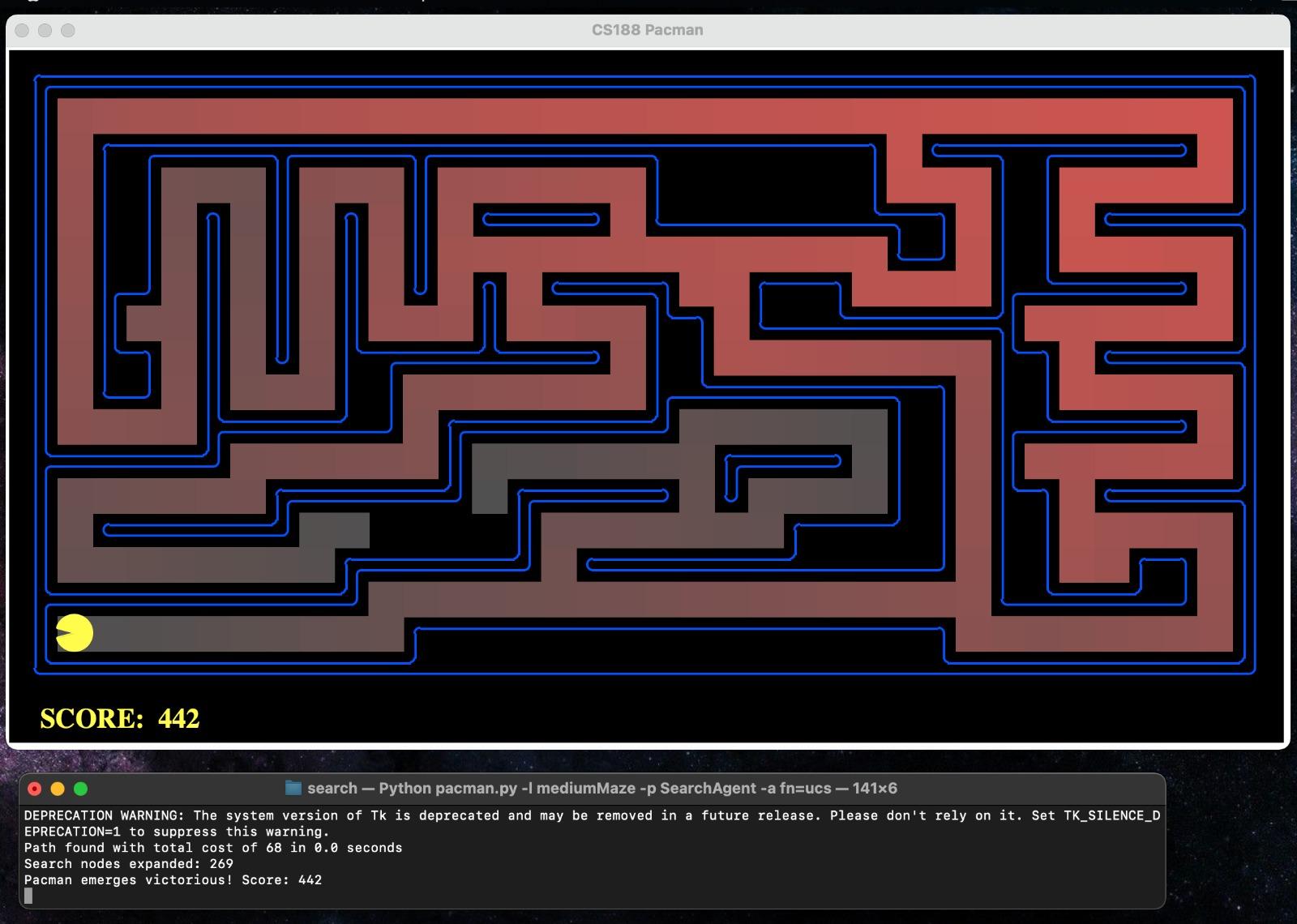


Q2. B.

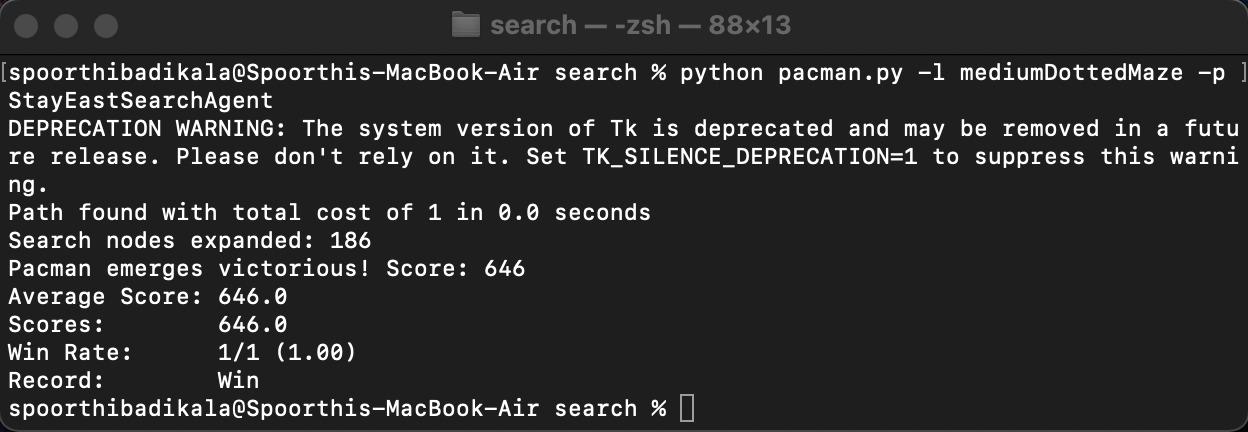


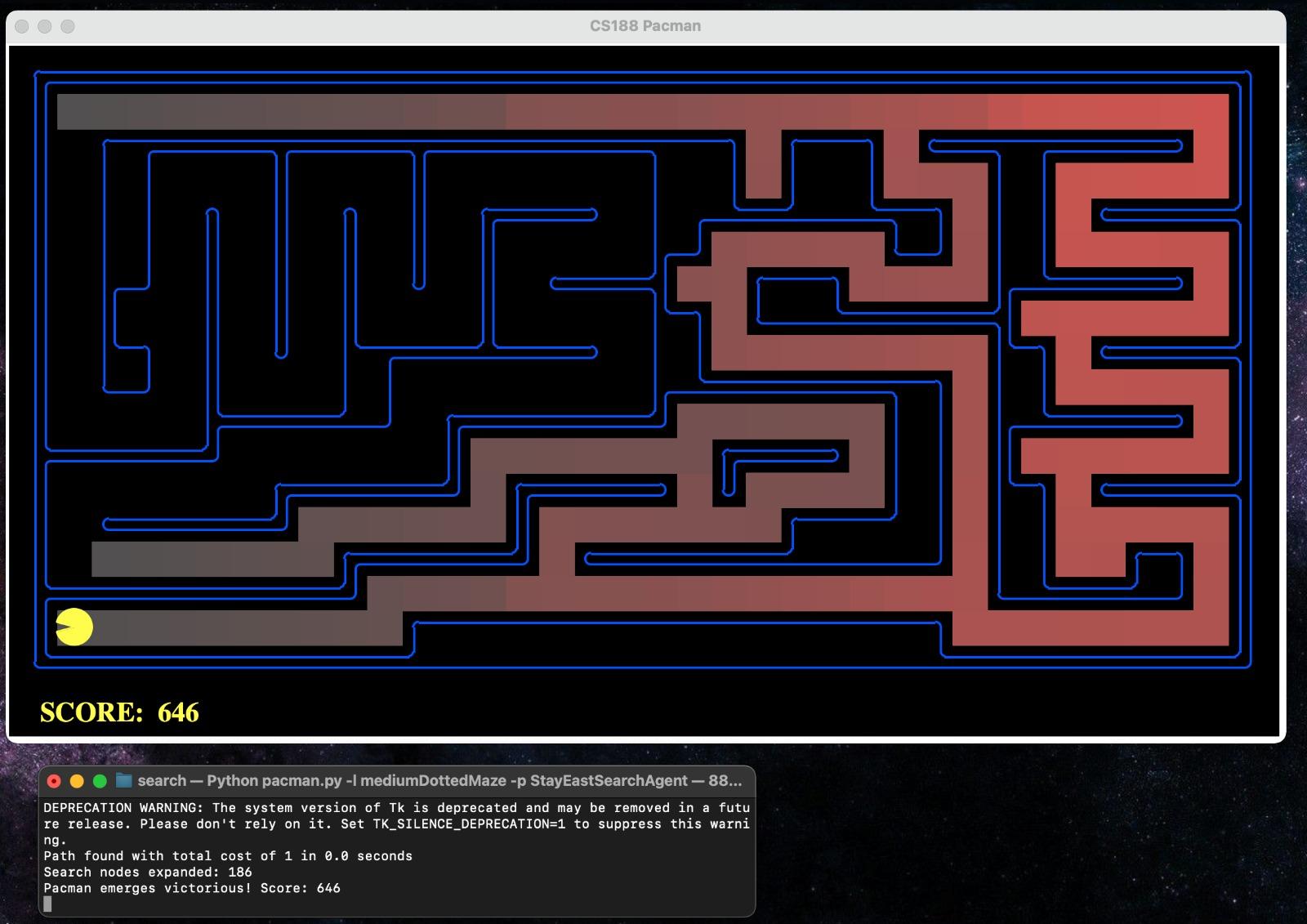
Q3. A.





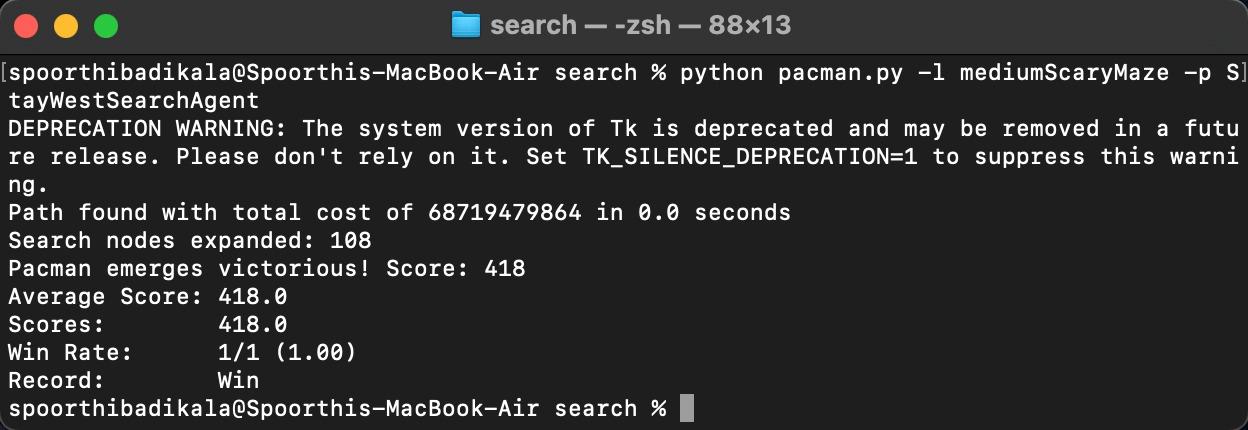
Q3. B.





Q3. C.





Q4.

