Artificial Intelligence

Wendi He 贺文迪 517030910315

1.Q1

The maze problem can be framed as aa state-space search problem, where every little square whose value is not 1 will be regarded as a node in a graph and if any two squares who have common edges will be added an edge connecting the two nodes on the graph. In this case, we derive a graph to represent the maze.

Now, the node who has the value 2 and 3 are the stating node and the ending node correspondingly. And the other nodes all have the value 1. The cost for every edge is 1. Our goal is to find a path from a path from the stating node to the ending node with the shortest path, which means the smallest cost in other words.

2.Q2

There are several searching algorithms that we have learnt in class. To solve the optimal solution of the maze problem, here we can use are BFS searching and backtrack searching.

BFS: The main idea of the Bread First Searching (BFS) is to explore all nodes in order of increasing depth. BFS maintains a queue of states to be explored. It pops a state off the queue, then pushes its successors back on the queue. In every state, it will take d actions and the solution had d action. Since we only need to search b^d nodes, the time complexity is O(b^d). However, the space complexity is also O(b^d).

Backtrack: The backtrack algorithm is called recursively on the current state s and the path leading up to that state. If the goal has been reached, then we can update the minimum cost path with the current path. Otherwise, we consider all possible actions a from state s, and recursively search each of the possibilities. To maintain the stack for the recurrence, the space complexity is O(D). Since the algorithm need to search for every nodes on the searching tree, so the time complexity is O(b^D).

3.Q3

The detail of the algorithms can be seen in the python file. The reason that I choose the BFS searching because this algorithm has a low time complexity O(b^d), which means it can solve the problem in a shorter time. Besides, its space complexity is acceptable and also it is very easy and direct to realize in python codes.