Lab01: Search Strategies

Problem description

A robot has been sent to a maze of size $N \times N$ and it must navigate the maze to reach the exit. Given that the robot has the map of the maze and it knows that its initial location is at square 0 and needs to reach some square E (from 0 to $N^2 - 1$) to go out of the maze. Figures 1 demonstrates a maze of size 8×8 and the exit E is at square 61.

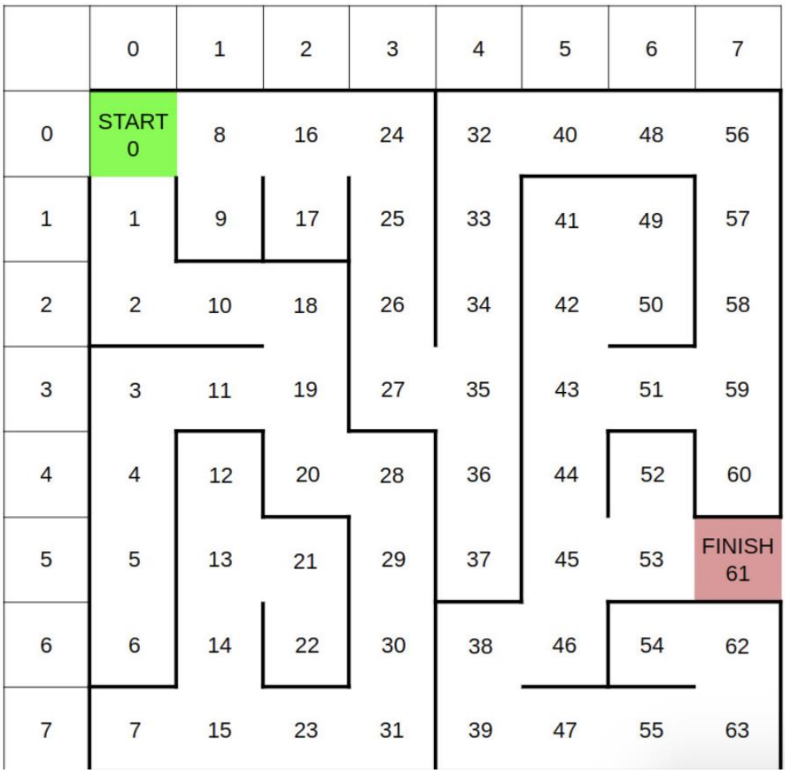

Figure 1. An example maze.

Let us use our AI knowledge to help the robot search its path out of the maze. The robot must obey the following rules for searching the maze (which are hardcoded in its memory)

- The (x, y) coordinates of each node are defined by the column and the row (shown at the top and left of the maze in Figure 1, respectively). For example, node 13 has (x, y) coordinates (1, 5).
- Process neighbors in increasing order. For example, if processing the neighbors of node 13, first process 12, then 14, then 21.
- Use a priority queue for your frontier. Add tuples of (priority, node) to the frontier. For example, when performing UCS and processing node 13, add (15, 12) to the frontier, then (15, 14), then (15, 21), where 15 is the distance (or cost) to each node. When performing A*, use the cost plus the heuristic distance as the priority.
- When removing nodes from the frontier (or popping off the queue), break ties by taking the node that comes first lexicographically. For example, if deciding between (15, 12), (15, 14) and (15, 21) from above, choose (15, 12) first (because 12 < 14 < 21).

- A node is considered visited when it is removed from the frontier (or popped off the queue).
- You can only move horizontally and vertically (not diagonally).
- It takes 1 minute to explore a single node. The time to escape the maze will be the sum of all nodes explored, not just the length of the final path.
- All edges have cost 1.
- The exit can be at any square and the maze is not guaranteed to be escapable.

Problem representation

Input: The maze is represented in a text file as follows.

- The first line includes a positive integer N only, which indicates the size of the maze.
- Each of the next $N \times N$ lines contains an adjacency list of node i. Nodes in the list are unordered and separated by white spaces.
- The last line includes a non-negative integer E (from 0 to $N^2 - 1$), which indicates the exit.

For example, with the maze shown above, a part of the input file will be as follows

Input file content	Note
8	The size of the maze, $N \times N$
8 1	Node 0's adjacency list
10 1	Node 1's adjacency list
11 4	
……	……
53	Node 61's adjacency list
54 63	Node 62's adjacency list
55 62	Node 63's adjacency list
61	The exit node

Output: Print to the console the following information

- The time to escape the maze
- The list of explored nodes (in correct order)
- The list of nodes on the path found (in correct order).

Search strategies implementation

You are asked to implement the following search strategies and provide their output results.

- *Breadth-first search*
- *Uniform-cost search*
- *Iterative deepening search* that uses depth-first tree search as core component and avoids loops by checking a new node against the current path
- *Greedy-best first search* using the Manhattan distance as heuristic
- *Tree-search A** using the same heuristic as above

For consistency, the goal node (i.e., the exit) will be included to every list of explored nodes (though not all strategies really put it to the frontier).

Prepare a function for each strategy and name the function with a reminiscent name.

Sources of AI search strategies are widely available, yet it is highly prohibited to copy them at any level of significance. You may be subjected to severe penalties for any signal of plagiarism.

References

Original problem description:

http://ashishgupta.me/articles/2019-05-02-AI-Search-Algorithms-Implementations/

Rubric

Criteria	Points
Implement the search strategies	2.5 (0.5pt each)
Correct list of explored nodes (in correct order)	5 (1pt each)
Correct list of nodes on the path found (in correct order)	2.5 (0.5pt each)

Submission preparation

This is an INDIVIDUAL assignment.
Prepare a folder that includes the following subfolders

- SOURCE: all Python files should be put in here
- INPUT: some example mazes in addition to the given example
- OUTPUT: corresponding search results to the mazes in the INPUT folder
- DOCUMENT: a PDF-formatted file that presents a check list of what you have/have not done and a brief description of main functions (so that the Lab Instructors do not miss any of your efforts)

Name the main folder following your Student ID and compress it in common format.