Personal Code Assignment

A. Requirements

Code (80%)

In this assignment, you need to complete A* search in Sliding Puzzle Problem using Python, Java, C, or C++. Your code must be a complete executable program instead of only a function. We guarantee test data strictly compliance with the requirements in the description. Directly copying others' codes are not allowed (either online or from your classmates). <u>Once I find codes that are overly similar (determined by code-checking software), both students will get zero in this assignment.</u>

Libraries in this assignment:

- For C/C++, you can only include standard library.
- For Java, you can only import java.util.
- For Python, you can only import standard library. You cannot import libraries such as numpy.

Report (20%)

You also need to write a report in .pdf type to contain the followings:

- 1. The pseudocode of your implementation, including how you implement A* search algorithm and apply it into this application.
- 2. What problems have you encountered during your implementation. You need to post the printing results when you encountered bugs, and explaining how you ultimately resolved them.

Please note that the report is for proving the originality of your codes. Therefore, showing above mentioned points are critical. No specific format requirements in report writing.

Marking Standard

For the same problem, we provide an easier version (A, 40 points and a harder version (B, 80 points. Version A requires only one output, and version B requires two outputs. For version B, you need to make sure two outputs are both correct to pass the test.

We will test your code with 100 test samples and mark with your testing accuracy (accuracy (%) = #correct_output / #test_samples). The scores of the outputs and the report will not affect each other. <u>The</u> marking standard of them are:

- 1. Scores of version A (40%): 40 points * accuracy
- 2. Scores of version B (80%): 80 points * accuracy
- 3. Scores of the report (20%): each required content values 10 points.
- 4. <u>Final Scores (100%)</u>: max(Scores of version A, Scores of version B) + Scores of the report.

B. Sliding Puzzle Problem

Description

In a 2 x 3 board, there are 5 tiles represented by the numbers 1 to 5, and one empty space represented by 0. A move is defined as swapping the 0 with an adjacent number (left, up, right, or down). The puzzle is solved when the board's configuration is [[1,2,3], [4,5,0]].

Given an initial state of the board, print the minimum number of moves required to solve the puzzle and the moving order. If the puzzle is unsolvable, print -1 and None.

Requirements

- 1. There is one input <u>list</u> named board to represent the statue of the board.
- 2. There are two required outputs:
 - One <u>integer</u> named steps to represent the minimum number of moves required to solve the puzzle.
 If the puzzle is unsolvable, print -1. <u>Required in both version A and version B.</u>
 - 2) One <u>string</u> named moving to represent the movement track of 0 to achieve the minimum number of moves. For each movement, there are four possible directions. The hierarchy of movement directions are fixed that left (denoted as '1') has the highest priority, followed by up ('u'), then right ('r'), and finally down ('d') being the lowest in priority. If the puzzle is unsolvable, print None. <u>Required in version B.</u>
- 3. You need to implement this with A* search. Otherwise, you will receive zero in this assignment. I also wrote some supporting functions (in Python) in utils.py and you can import the file to simplify your implementation.

Input and Output Format

The input of test samples will be string. Therefore, you need to first transform the input into list. I wrote some functions (in Python) for you to simplify the problem solving, including the input format transformation. Please use input() to obtain the input string and import utils to use these supporting functions in Python.

For version A, you only need to print(steps) to output your answer.

For version B, you need to print(steps) and print(moving) to output your answers.

Samples

Sample 1

1	2	3
4		5

Input: board = [[1,2,3],[4,0,5]] **Output:** steps = 1, moving = 'r' **Explanation:** Exchange 0 and 5. One step.

Sample 2

1	2	3
5	4	

Input: board = [[1,2,3],[5,4,0]] Output: steps = -1, moving = " Explanation: Unable to achieve the goal.

Sample 3



Input: board = [[4,1,2],[5,0,3]] Output: steps = 5, moving = 'lurrd' Explanation: Not moving: [[4,1,2],[5,0,3]] Moving left ('I'): [[4,1,2],[0,5,3]] Moving up ('u'): [[0,1,2],[4,5,3]] Moving right ('r'): [[1,0,2],[4,5,3]] Moving right ('r'): [[1,2,0],[4,5,3]] Moving down ('d'): [[1,2,3],[4,5,0]] In version A, the sample inputs and corresponding outputs with string format will be:

Sample Input (format: string)	Sample Output (format: string)
[[1,2,3],[4,0,5]]	1
[[1,2,3],[5,4,0]]	-1
[[4,1,2],[5,0,3]]	5

In version B, the sample inputs and corresponding outputs with string format will be:

Sample Input (format: string)	Sample Output (format: string)
[[1,2,3],[4,0,5]]	1
	r
[[1,2,3],[5,4,0]]	-1
	None
[[4,1,2],[5,0,3]]	5
	lurrd

C. Submission

After finishing this assignment, you are required to submit your code to the Online Judge System (OJ) and upload your .zip package of your code files and report to BlackBoard.

Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (<u>http://10.26.200.14/d/csc3180_2024_spring/homework/65b9169d18de8e8c191be562</u>, campus only) to gain marks for the code part. You can submit your solution of one problem for no more than 80 times.

After you have submitted your program, OJ will test your program on all test cases and give you a grade. The grade of your latest submission will be regarded as the final grade of the problem. The problem is tested on 100 test cases, and your scores will only be determined by your testing accuracy within the required time and memory.

Note: If you are new to OJ system, you need to reset your account (not register). You can select 'language' on the bottom of the page.

BlackBoard

You are required to upload your source codes and report to the BlackBoard platform. You need

to name your files with following rules and compress them into AS_<Student ID>.zip :

AS_<Student ID>.zip |-- AS_ <Student ID>.java/py/c/cpp |-- AS_Report_<Student ID>.pdf

Late Submissions

Submissions after \underline{Week} 7 would be considered as LATE.

The submission time is the latest submission time in OJ and BlackBoard.

There will be penalties for late submission:

- 0-24 hours after deadline: final score = your score $\times 0.8$
- 24–72 hours after deadline: final score = your score $\times 0.5$
- 72+ hours after deadline: final score = your score $\times 0$

D. Others

If you have questions of OJ system, please check:

OJ wiki (http://10.26.200.14/d/csc3180_2024_spring/wiki/help)

If you have questions for the problem, please contact:

Chaoxun Guo (222010062@link.cuhk.edu.cn)