
CSIS, BITS Pilani K. K. Birla Goa Campus
Artificial Intelligence (CS F407)

Programming Assignment 1

Total Marks: 15

Submission Deadline: 9 PM on 28/09/2020 (Monday)

Each student must individually do this programming assignment. Your program must be written in Python and should run (without errors) on Python 3.6 or later.

Submitting others' program as your own (plagiarism) will result in -5 marks being awarded. Note that the deadline is **9 PM** and not midnight. Five marks per day will be deducted for submissions after the deadline. It will be your responsibility to submit the assignment well in advance and avoid unforeseen problems like power failures etc.

Question 1 (15 marks)

The figure below shows the 15-puzzle problem. The square marked as 0 is the empty space. The remaining characters (i.e. 1 to F) represent the 15 blocks that can be slid into the empty space if they are adjacent. You can assume that the initial state is such that it is possible to reach the Goal state shown below. The possible actions are Up, Down, Left and Right.

4	2	6	3
8	1	7	B
C	5	0	F
D	9	A	E

(a) A possible initial State

0	1	2	3
4	5	6	7
8	9	A	B
C	D	E	F

(b) Goal state

Write a python program to find the minimum path cost (assume unit step cost) to go from the initial state to the Goal state. For the optimal solution, you must show the sequence of actions to go from the initial state to the goal state. A sample program is given ("ROLLXYZ.FIRSTNAME.py"). Run the program and see the output. You must modify this program. Note: While evaluating your program, a different initial state (i.e. initial_state4.txt file) will be used.

- (a) If you have followed all the instructions and your code runs on Python 3.6 or later without any errors, then you will receive 2 marks.

- (b) In addition to point (a), if your program gives a (correct) solution in less than 30 seconds, then you will receive 2 more marks. This solution need not be optimal.
- (c) In addition to points (a) and (b), if your program also gives an optimal solution in less than 30 seconds, then you will receive 5 more marks. To make it more clear, if you have followed all the instructions and your program is able to find an optimal solution in less than 30 seconds, then you will get 9/15 marks ($2 + 2 + 5$).
- (d) In addition to points (a), (b) and (c), if your program gives an optimal solution in a time which is less than $\text{Mean} - \text{SD}$, then you will get 3 more marks. Here, Mean refers to the mean running time of all the student programs and SD refers to the standard deviation. Note that the Mean and SD will be calculated using **only** those programs that give an optimal solution within 30 seconds. You must report the number of nodes generated while finding the optimal solution. The aim should be to minimize the number nodes (that are generated) so that the optimal solution is found in lesser time. To reiterate, you will get 12/15 marks if criteria (a), (b), (c) and (d) are satisfied.
- (e) In addition to points (a), (b) and (c), if your program gives an optimal solution in a time which is less than $\text{Mean} - 2 * \text{SD}$, then you will get 6 more marks. In this case, you will receive full marks (i.e. 15/15) for the assignment. Again, you must report the number of nodes that were generated while finding the optimal solution.

Instructions for submission

- You must modify “ROLLXYZ_FIRSTNAME.py” file given in the assignment folder.
- A different “initial_state4.txt” file will be used during evaluation. But the difficulty level will be similar to the problem in “initial_state4.txt” file.
- All your code must be written in the same program file (i.e. “ROLLXYZ_FIRSTNAME.py”).
- Run the given program (i.e. “ROLLXYZ_FIRSTNAME.py”) and see the output.
- A python implementation of the priority queue is included. See “heapq_test.py” file.
- Optimal path costs are given for four problems (see “solutions.txt” file). First try to solve the easiest problem.
- Carefully read the comments in “ROLLXYZ_FIRSTNAME.py” file. Strictly follow the instructions in the comments.
- You may use an additional file “ROLLXYZ_FIRSTNAME.dat” (note the file extension). This can be used to store some information that you feel might be useful. Using this additional file is **not** mandatory.
- The size of “ROLLXYZ_FIRSTNAME.dat” file (if it is used) should not exceed 640 kB (kilo-byte).

- The size of your program file “ROLLXYZ_FIRSTNAME.py” should not exceed 60 kB (kilobyte).
- Include a two slide presentation as a PDF file. First slide must describe your approach. Second slide must describe why your approach will always give an optimal solution. The description on each slide must be less than 150 words. The PDF file must be of the simplest possible format (no images), and its size should not exceed 60 kB (kilobyte).
- Your submission should contain just three files : ROLLXYZ_FIRSTNAME.dat, ROLLXYZ_FIRSTNAME.py and ROLLXYZ_FIRSTNAME.pdf. You can zip the folder containing the three files. Name of the folder should again be ROLLXYZ_FIRSTNAME. Don't submit any other file. File size limits mentioned above are for uncompressed files.
- You should modify the filenames to your own roll number and first name. Use capital letters in roll number.
- Run your program on another system (if available) to ensure that it does not give any error.
- Marks will be deducted if instructions are not followed.
- I recommend that first you implement an algorithm which will always give an optimal solution (first program). You can then think of ways to improve this algorithm. You can explore. You can submit the first program as soon as it is done. Improvements (if any) can be submitted later. When submitting improvements, you must be sure that it will always give an optimal solution; otherwise, you may lose marks. When you upload a new submission, delete the old submission. Assignment submission should be completed before the deadline.
- Contact the IC if your program is taking more than 30 seconds (on your PC/laptop) for the given problems.