CS3642 Artificial Intelligence

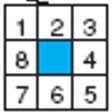
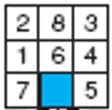
Programming Assignment #1

To implement Breadth-First Search (BFS), and Iterative-Depth-First Search (IDFS) algorithm, to solve the 8- puzzle problems (i.e. find the goal).

# 8-Puzzle Problem:

8-puzzle is a problem where you have total 9 blocks as shown in the figure below. 8-blocks/cells are marked with a number from 1 to 8 and one cell is blank/empty, the goal is to organize numbered block in clockwise in an ascending order (shown in the final state).

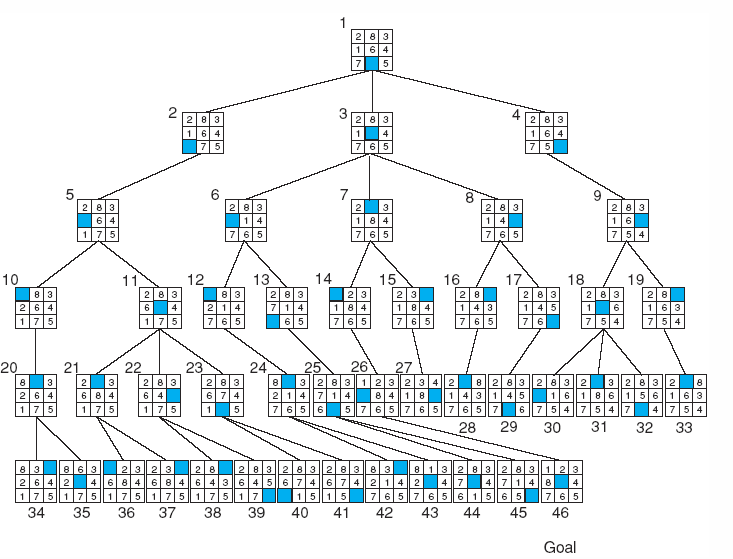
Given an initial state and final state below. Write an BFS/DFS algorithm to solve the following 8- puzzle problem. Your BFS/DFS algorithm should create all intermediate states until it reaches the final states. Your code should be executable and print all the states while executing. Examples of such BFS algorithm is shown below.



Initial state (vertex)

Final goal state/vertex

The program is to start from an initial configuration to find the goal configuration. A solution to the problem is an appropriate sequence of moves, such as “move tiles 5 to the right, move tile 7 to the left, move tile 6 to the down, etc”. You should always start with an initial random state **(do not hard-coded in your program). Your program should be able to deal with any initial configuration.**



***Note for DFS: You need to make sure that you do not re-add any of the parents which are already in the tree branch. Otherwise, you will create a loop.***

# You must write your own codes for the algorithms. Make sure your submission meets all of the requirements and free of plagiarism.

You may write your code in a contemporary language of your choice; typical languages would include C/C++, Python, Java, Ada, Pascal, Smalltalk, Lisp, and Prolog. A GUI interface is preferred.

**Python notebook:** If you are using python, I would recommend using jupyter notebook/lab. So that I can see the visible output of your source code. the cell outputs should be visible with a sample test example if applicable. Then create a demo video presentation of your source execution showing the intermediate states and submit this demo with your source.

In case of you are using python scripts instead of notebook, then create a demo video presentation of your source execution showing the intermediate states and submit this demo with your source.

**Other user:** Who uses other language, should create a demo video presentation of your source execution showing the intermediate states and submit this demo with your source.

**Submission**

***Do not submit zip file.***

Please submit your deliverables to D2L Assignments folder: Programing assignment 1 (PA1). You need to submit 4 different things:

1. Your source code saved as yourlastname.
2. Your source code saved as yourlastname.txt - copy and paste your entire source code, save as .txt
3. Your report yourlastname\_Report.doc as a word document
4. A short video demo: (3-5 minutes) You can send a link to your video if you could not upload to D2L
   1. Explaining your code
   2. Run your code, showing the intermediate states and final graph

Once you submit, D2L will perform a similarity check for your submission and show you the result. Your similarity score on your code as txt file must be lower than 50% unless something essential is described in the report. *Otherwise, 5 points will be deducted for every additional 5% similarity*. For example, you would get a maximum 95 if your similarity is between 51% and 55%.

# Project Report:

The report does not have any specific format. Three things you must include:

1. What you did for the project, i.e., explain the project with your own word.
2. Explain your program/code.
3. Explain output, include screen shot from your run.