BIM309 - Homework I

The 8-puzzle problem is a puzzle invented and popularized by Noyes Palmer Chapman in the 1870s. It is played on a 3-by-3 grid with 8 square blocks labeled 1 through 8 and a blank square. The goal of the puzzle is to rearrange the blocks so that they are in order. You are permitted to slide blocks horizontally or vertically into the blank square. The following shows a sequence of legal moves from an initial board position (left) to the goal position (right).

initial

	1	3
4	2	5
7	8	6

=>

1		3
4	2	5
7	8	6

=>

1	2	3
4		5
7	8	6

=>

1	2	3
4	5	
7	8	6

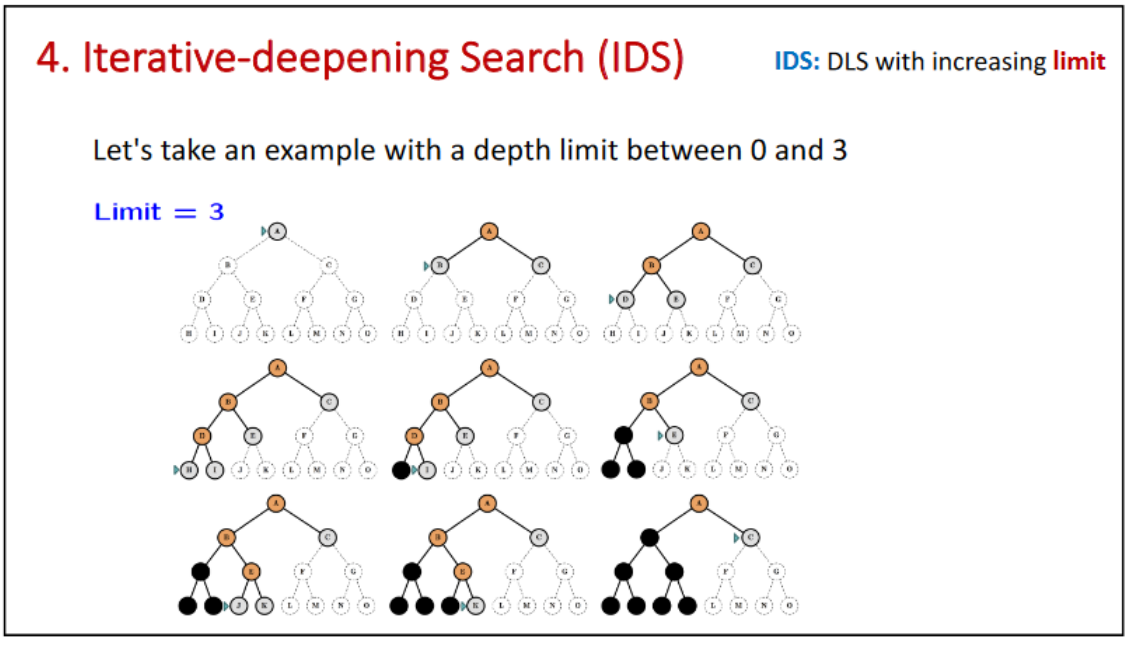
=>

goal

1	2	3
4	5	6
7	8	

Your task is to complete given code in order to build a 8-puzzle solver utilizing "Iterative-deepening search". **Optimal solutions should be found within a minimum number of moves.**

Recall from lecture notes



Here is another example for searching next states. In order to find a utilized solution and avoid infinite loop, graph shouldn't enqueue the neighbor if its board position is the same as the previous state.

previous

1	2	3
8		4

=>

state

1		3
8	2	4

=>

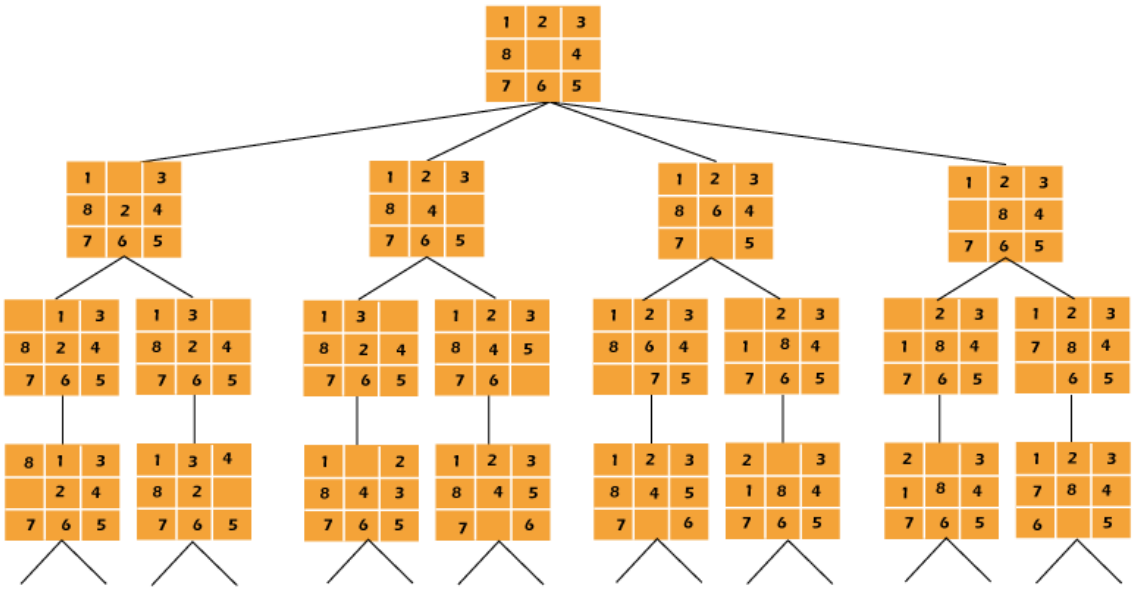
disallowed

1	2	3
8		4

7	6	5

7	6	5

7	6	5



- Modify the script file (HomeworkI.ipynb) provided for you to meet the requirements described in the file. Your solution can be a “py” file.
- Submitted file should be an archive (tar, zip, rar, etc.) named after your id number (e.g., 12345678910.zip).
- Submit your own work.

“Honesty is the best policy; I will stick to that.” – Miguel de Cervantes