



Ghulam Ishaq Khan Institute of Engineering Sciences & Technology

Faculty of Computer Science & Engineering (FCSE)

CS231B – Discrete Mathematics



Assignment 2 (Fall 2019)

Total marks – 06 points Due Date – 13 December 2019 Instructor – Dr. Raja Hashim Ali

Instructions:

1. There is **1** part of this assignment, carrying **6** marks.
2. This assignment is an individual assignment, please do not share your solutions, designs or ideas with other students.
3. You have to build a project repository on git, then add each document you make and then add the project files. The git repository must contain the complete project history including all commits.
4. Send me an invitation to be administrator on your git repository, so that I can view your progress. Remember No GIT and no invitation to me, no marks for this assignment.
5. Please remember to commit your code on the repository periodically, so that there is always backup available.
6. Read the details carefully first, understand the statement, write your algorithm in pseudocode and then start coding.
7. There will be vivas in the fifteenth week of selected students.
8. Passing this assignment is a must! You will not be allowed to sit in final exams unless you clear this assignment.
9. Any verified case of cheating (including sharing ideas or solutions) will lead to straight F in this course, (No ifs and Buts).
10. You can use internet, TA or me to get help.

CLO Mapping Table:

Parts of Assignment	
1	

Marks Sheet:

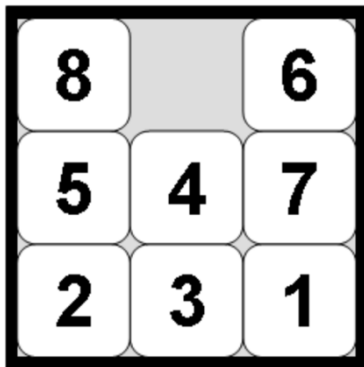
Parts of Assignment	
1	6 marks

Dated: 19 November 2019

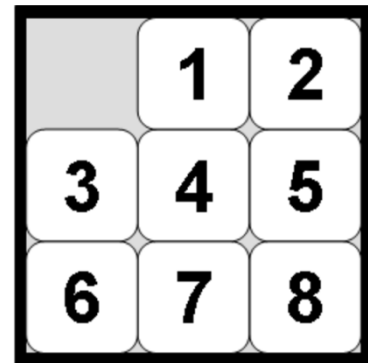
Instructor: Dr. Raja Hashim Ali

Searching for a solution using graph exploration techniques

Consider the 8-puzzle problem, where the following states are start (initial) state and final end (goal) state respectively. Note that you must use graph exploration techniques (a state once explored should not be explored again using a set of used states). Then determine the path taken to reach from the given start state to reach the final end state using **breadth first** technique. As example, an initial state is given below but the user may enter an initial state of his own choice. But the final state will not change or should be hard coded to the following.



Initial State



Final State

The user will input the initial state from command line/file. The maximum allowed moves are four, where only 'tiles' (numbers) can be moved to the empty space in either vertical (upward or downward) or horizontal (left or right). Please note that only the adjacent tiles to the empty tile can be moved and no tile can replace another tile. Moving a tile in diagonal is not allowed! For example, the only legal moves from initial state are

- 1) 8 moves to the right, or
- 2) 4 moves to the top, or
- 3) 6 moves to the left.

Also, if a move leads to a visited state, that move is not allowed (as graphs do not allow exploration of already visited states). For example, if we had moved 8 to the right as first move, moving 8 to the left as second move would lead back to the initial state; hence moving 8 to the left as second move is not allowed!

Once, a move has been taken, store both the move (for keeping track of the path) and the new state as visited state (to not search the visited states again).

Input: The initial state from the user (as either file or from command line in a format of your choice).

Output: The sequence of moves (path) that the user can replicate to reach the goal state from initial state.

Submissions: Link to your private git repository, containing the source code and a report that contains how I can run your program.