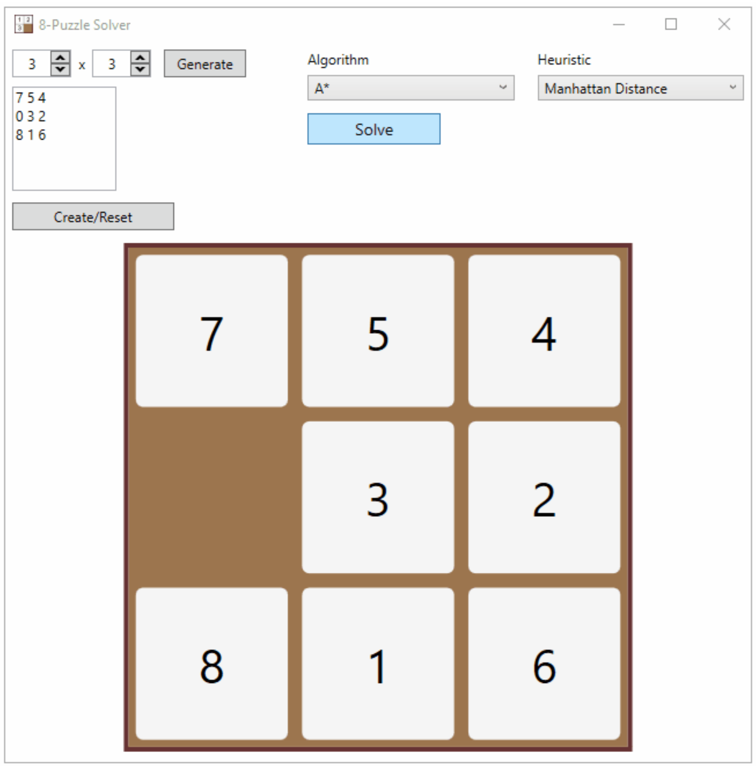
Implementation and software

Project#1 (10 points)



GiffingTool.com

Project -1 Description

- Find all the possible states of the 8-Puzzle starting from the given initial state. Note that, the states should be unique (no repetitions).
- Use the brute force search algorithm (BFS) to find the path to reach the goal state using the possible states of 8-puzzle.
- You can use Matlab/C++/Python for programing

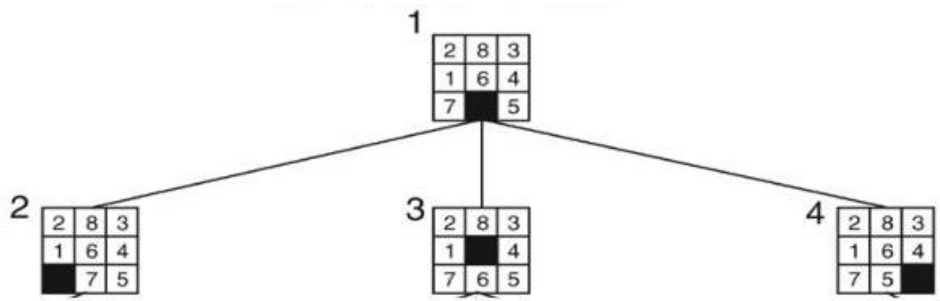
Example

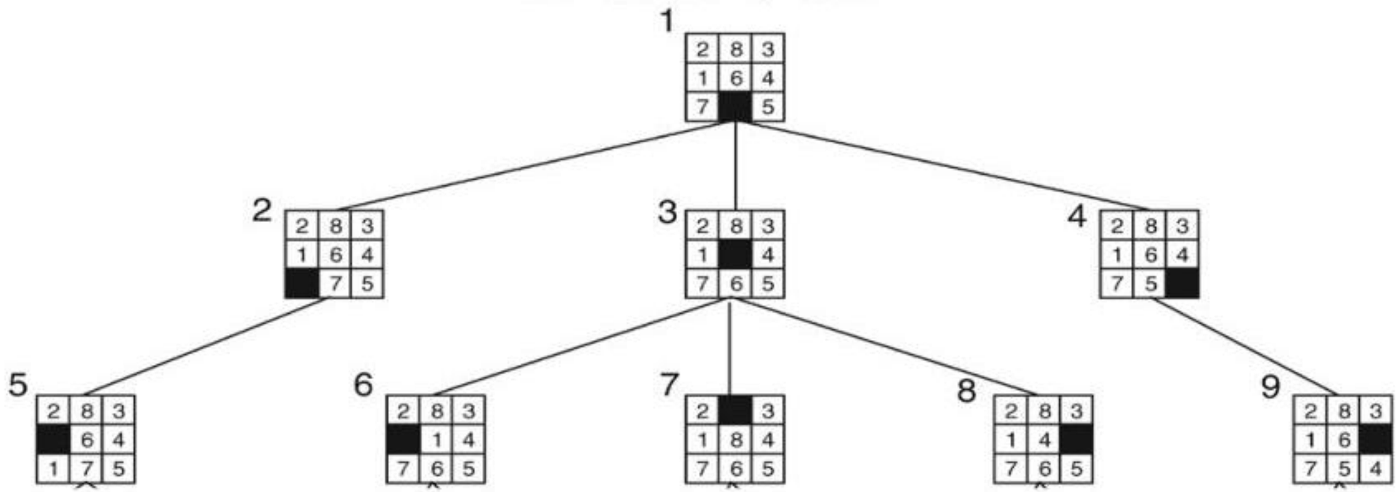
- Following example shows different configurations of the 8-puzzle generated from the initial state.
- From the initial state of the puzzle, use different moves in all the directions to generate new states, check the validity of the newly generated node.

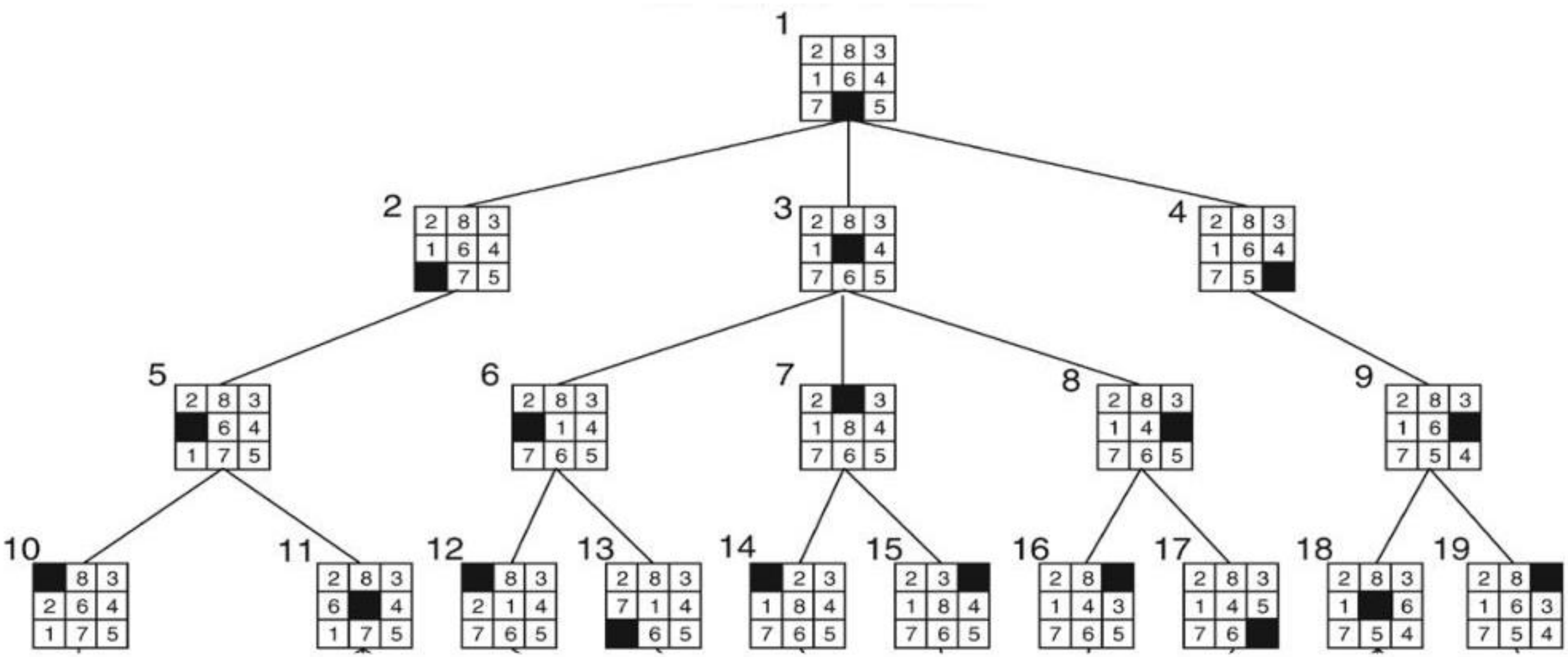
First method

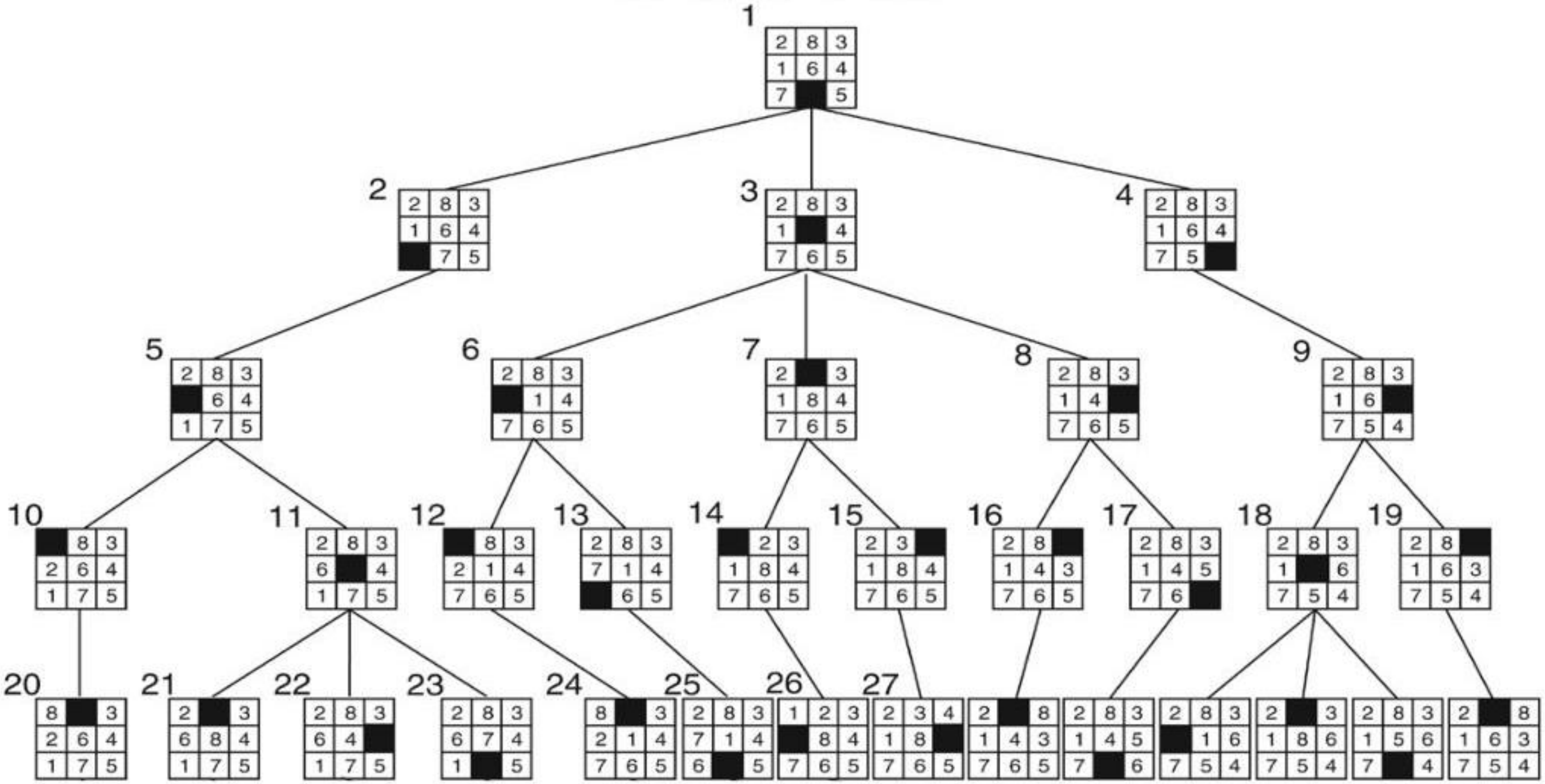
1

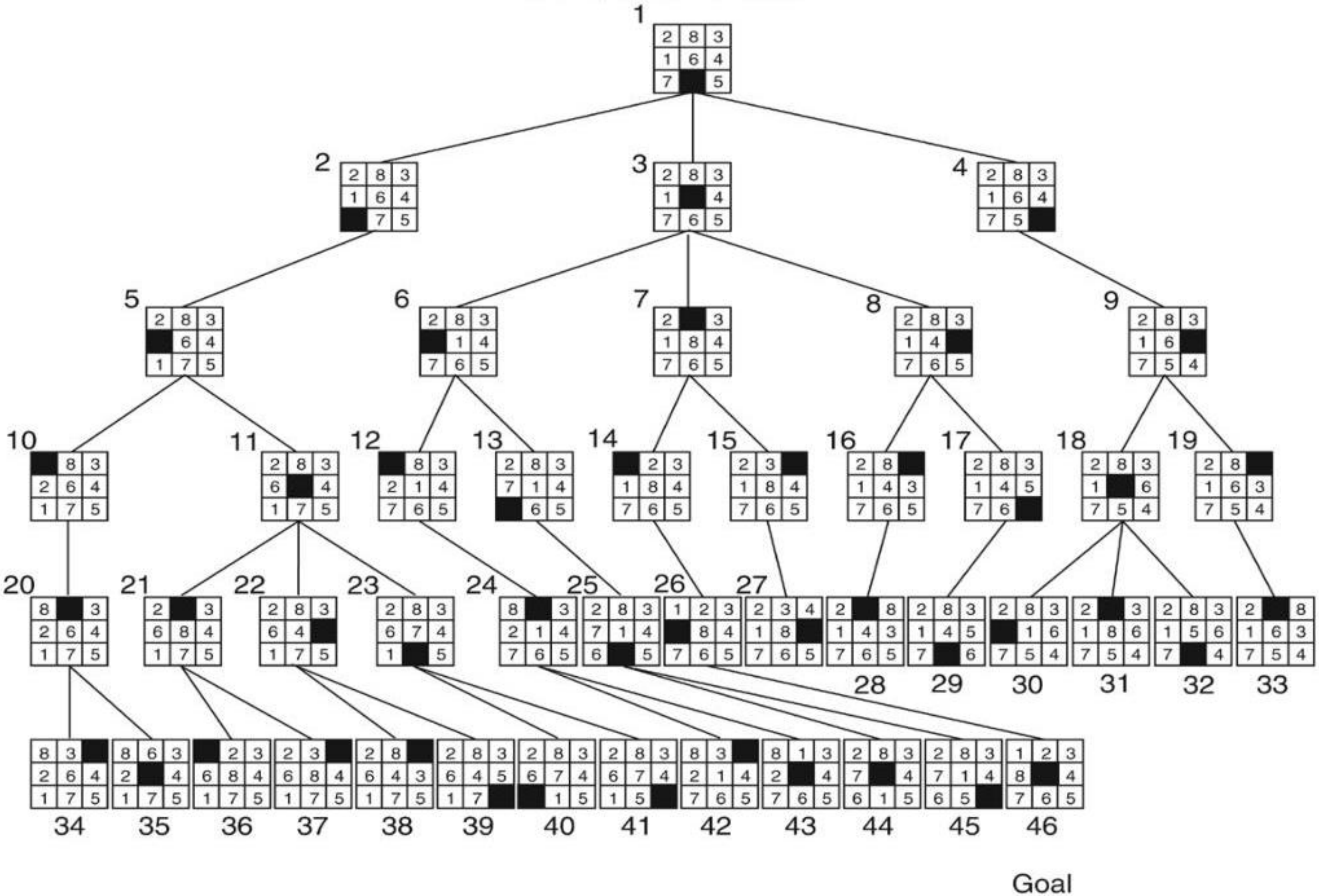
2	8	3
1	6	4
7		5


1
2
3
4


1
2 8 3
1 6 4
7 5
2
2 8 3
1 6 4
7 5
3
2 8 3
1 4
7 6 5
4
2 8 3
1 6 4
7 5
5
2 8 3
6 4
1 7 5
6
2 8 3
1 4
7 6 5
7
2 3
1 8 4
7 6 5
8
2 8 3
1 4
7 6 5
9
2 8 3
1 6
7 5 4


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19


1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27


Goal

Brute Force search algorithm

- In computer science, **brute-force search** or **exhaustive search**, also known as **generate and test**, is a very general problem-solving technique and algorithmic paradigm that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.
- As you find different states of the 8-puzzle, you will be checking for the goal state.
- Once you find the goal state, use the node information to find the parent node. This backward process will finally lead to the initial state.
- This algorithm does not consider the cost (number of moves) to reach the goal state.

Ref. - https://en.wikipedia.org/wiki/Brute-force_search

MATLAB

Define Nodes matrix and NodesInfo matrix first:

```
Nodes= [];
NodesInfo=[];                          %   NodeInfo = [ Node #, Parent node #, Cost2Come (Number of Steps)]

Nodes_Init = [0 1 3; 4 2 5; 7 8 6];    % Define initial State of the puzzle, input from the user
                                       % For example: Nodes_Init= [0 1 3; 4 2 5; 7 8 6]
NodesInfo_Init = [1 0 0];              % Information matrix for the first node

Nodes_Goal = [1 2 3; 4 5 6; 7 8 0];    %   Assume [1 2 3; 4 5 6; 7 8 0] as the goal node



Nodes(:, :, 1) = Nodes_Init;       % Save initial state to nodes set
```

Sub_functions:

```
[X0 Y0] = BlankTileLocation(CurrentNode);              % Find the location of the blank tile
[Status, NewNode] = ActionMoveLeft(CurrentNode);       % Moves blank tile left, if possible
[Status, NewNode] = ActionMoveRight(CurrentNode);      % Moves blank tile right, if possible
[Status, NewNode] = ActionMoveUp(CurrentNode);         % Moves blank tile up, if possible
[Status, NewNode] = ActionMoveDown(CurrentNode);       % Moves blank tile down, if possible
[Nodes, NodesInfo] = AddNode(NewNode);                 % Function also checks whether node is new or not
[Moves] = Goal_Check(NewNode, Nodes_goal);             % Gives path from initial to goal state
```

Additional Points

- The MATLAB script to visualize the moves will be provided. To successfully run the script, output of your code must follow the given format.
  - Nodes is a 3-D matrix where the first two dimension represents individual nodes and the 3rd dimension represents the number of nodes explored.
  - NodesInfo is a 2-D or a 3-D matrix where the last dimension represents info of different nodes and other dimension gives the node number, parent node number, and the cost-to-come.
  - nodePath is a 3-D matrix as shown below:

    nodePath = cat(3, initial Node,………, goalNode)
  - sampleNode = [1 2 3; 4 5 6; 0 7 8];   % 0 represents the blank space

- Randomly 20-25 students will be selected who need to explain their code during TA office hours.

Due Date and Deliverables

- Due date: February 26 , 11:59 p.m
- Submit deliverables on Canvas
- Deliverables:
  – Source code
  – Output matrices : Nodes, NodesInfo and nodePath (the moves to reach the goal state)
  – A word file that explains how to run the program