Artificial Intelligence – Assignment 2 (Programming)

*Instructions:*

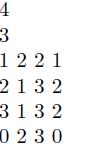
*Read the following problem and implement it using any language of your choice (C, C++ , Java or Python). Submit a Report on this problem (along with your codes) for each subpart.*

*Report should contain the heuristic used for the search, the number of nodes expanded in the state graph as well as describe the memory bounded algorithm implemented in the later part of assignment.*

*Deadline is 25 th August,2015.*

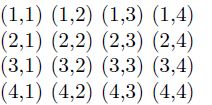
**Q1 a :**

Column Jump is a game played on a grid with balls of different colors. The initial board has some configuration of colored balls with at least one empty space. The objective is to remove all but one ball from the board. Balls are removed when they are jumped according to the following rules. If a ball of one color jumps over a different colored ball to an empty space, the jumped ball is removed from the board. Additionally, if multiple balls of the same color are in a line, they can be jumped and removed together (by a different colored ball, provided that an empty space is on the other side of the line). You can play the game at http://www.2flashgames.com/f/f-354.htm to get a sense of the rules. Be sure to play in Remover mode instead of Color Life mode. In this problem, we will consider a version of this game, and you will implement search algorithms to solve it. Our version can use grids of various sizes with different numbers of colors. You will be provided with several example input and output files for testing your implementation. The input format will be a text file. The first line of the file is the size of the (square) grid. The second line is the number of colors. This is followed by characters representing the colors in each location of the grid, with 0 representing an empty space. For example, an input file might look like this:

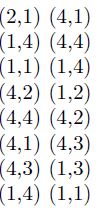


Your output should contain the series of moves required to solve the puzzle. There should be one move on each line, where a move is represented by two locations on the grid with (row, column) numbers.

The grid locations follow this convention:



So, the output file for the input given above looks like this:



Some possible admissible heuristics that you may use for this problem are

* The number of different colors on the board minus one.
* min(non-empty columns, non-empty rows)
* Number of lone balls (surrounded by spaces).

Implement A\* search to solve the Column Jump puzzle. Describe the heuristic you are using for A\*.

**Q1 b:**

Try to solve following test case with above implemented A\* search and then implement the memory bounded A\* to solve the following test case.

7

6

1 2 3 4 5 1 2

4 6 3 6 3 5 3

5 1 5 3 3 5 4

2 5 3 0 5 5 6

4 2 6 1 3 6 2

6 2 1 3 5 3 2

1 1 4 5 1 5 5