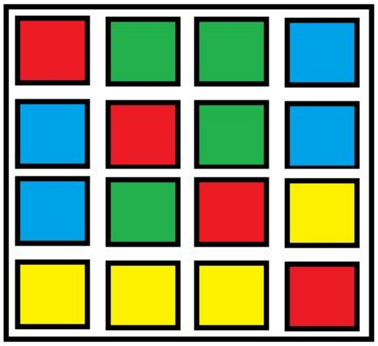
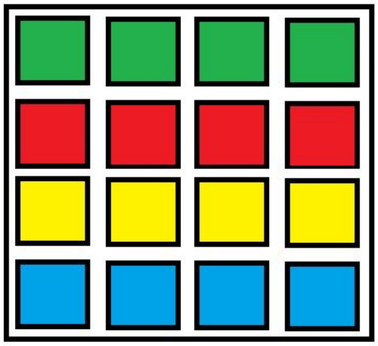
Offline for Section A1/B1

Submission Deadline: 6 October, 2017 11.55PM

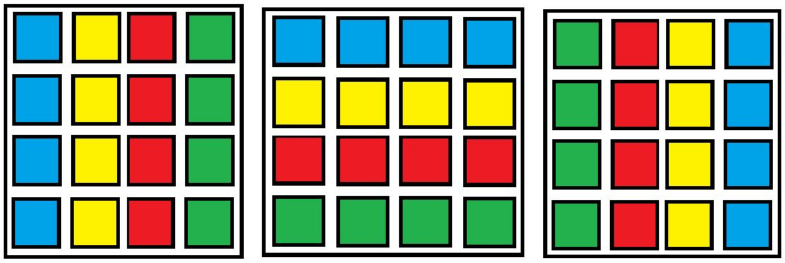
SOLVING “GRIDDLE- 2D Rubik’s cube” using SEARCH Techniques:

In this assignment you will have to solve a particular instance of the game “GRIDDLE” using search algorithm. The Problem 2D Rubik’s is not that much familiar with lots of people. The objective is to turn a board of multi colored squares into a oriented form using minimum of moves.


Initial State

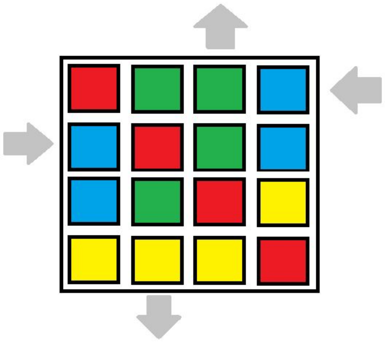

Goal State

There could be multiple possible goals state as you can be see from the rotation of the the figure



For each position you can apply four possible move, you can

1. Slide up a column
2. Slide down a column
3. Slide right a row
4. Slide left a row



It is easy to play, yet challenging, great fun and addictive.
The game can be played android app griddle.
https://play.google.com/store/apps/details?id=com.griddle

The tasks

1. You have to design heuristics to reach the goal state (any single/ any possible) of the problem using minimum number of moves. At least one of them should be admissible.

- Ultimately you have to do two things
    1. Design an algorithm to find minimum number of moves to reach into a fixed goal state
    2. Design an algorithm to find minimum number of moves to reach any goal state

2. You will have to implement A* search to solve at least one of the above cases
Slide: https://drive.google.com/open?id=0B-SybtAwa8vcUUZyNjJqcnhjbk0

3. You have to write a short report (1 page, hard copy), discussing the admissibility and the comparative performance (optimality and runtime) of your proposed heuristics.

Input File Format

The input file contains a series of inputs. First line contains the board size *n*. Next *n* line contains *n* integer each, i.e., the n*n board. The colors are represented by integers 1 to n. The file ends with a zero as board size.

The number of colors can be variable.

Sample goal State of the given scenario

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

Here 1, 2, 3, 4 represents GREEN, RED, YELLOW and BLUE colors respectively.

There for the above scenario the sample input will be

Sample Input:

```
4
2 1 1 4
4 2 1 4
4 1 2 3
3 3 3 2
```

And goal state can be

```
1 1 1 1
2 2 2 2
3 3 3 3
4 4 4 4
```

Or it’s different rotational form

<u>Output Format</u>

Output will be shown in console in the following format:

Number of moves: x
Next you have to print the moves and board state that you have performed to reach the goal state.

Example:

```
Row 1 left : print board
Row 3 right : print board
Column 4 up : print board
Column 1 down : print board
```

Implementing GUI is upto your choice and it’s appreciable

Evaluation Criteria:

To be updated