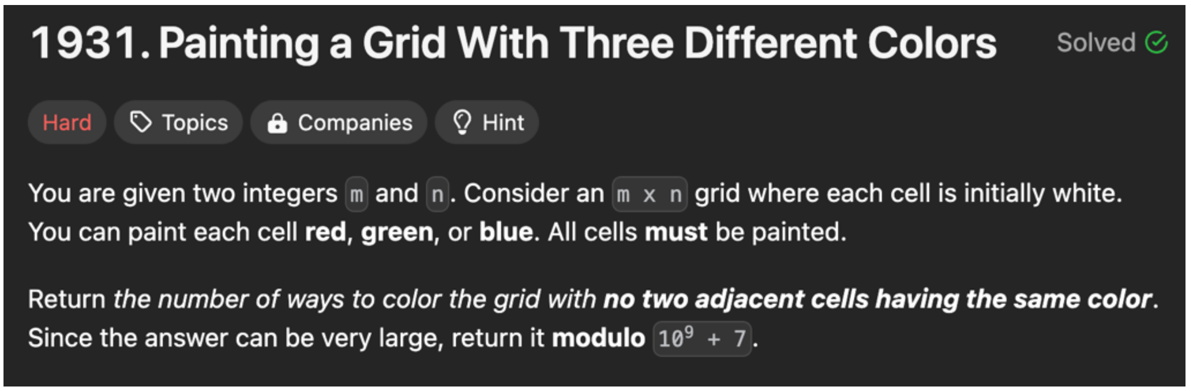
LeetCode 1931. Painting a Grid with Three different Colors (Hard)

May 18 2025

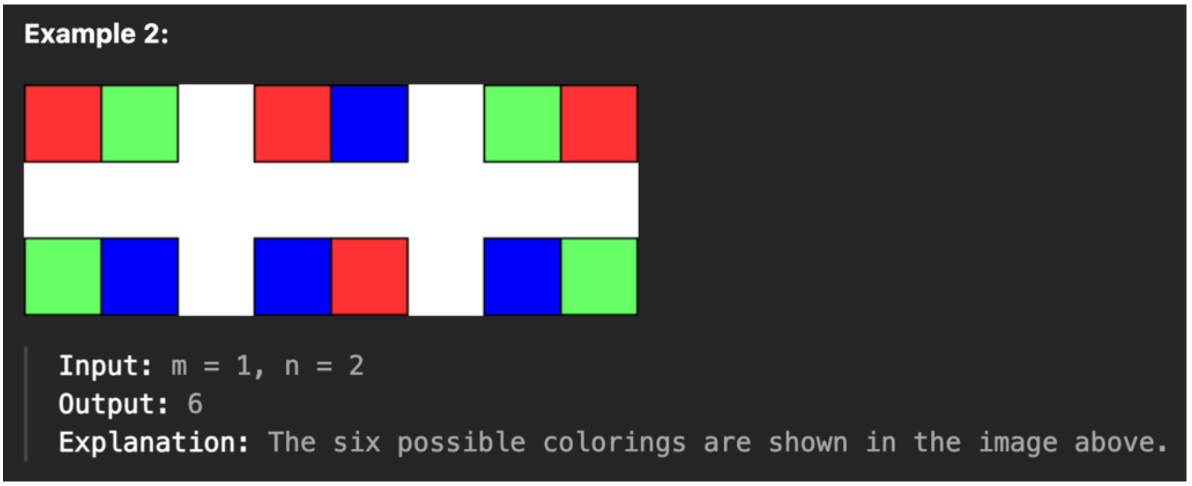


1931. Painting a Grid With Three Different Colors

Solved ✓

You are given two integers `m` and `n`. Consider an `m x n` grid where each cell is initially white. You can paint each cell **red**, **green**, or **blue**. All cells **must** be painted.

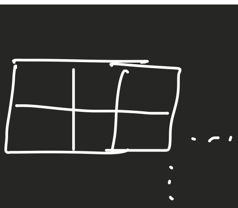
Return *the number of ways to color the grid with* ***no two adjacent cells having the same color***. Since the answer can be very large, return it **modulo** $10^9 + 7$.



Example 2:

```
Input: m = 1, n = 2
Output: 6
Explanation: The six possible colorings are shown in the image above.
```

Question: 1) Given a Grid of Squares.



2) You have 3 colors [🟥, 🟦, 🟩]

3) Task: # of ways to fill the grid so that no two adjacent squares (or cells) are of same color.

Thought Process:

→ I have choices to make. I can fill a cell any of the color. But there is a constraint that, the adjacent cells should have different color.

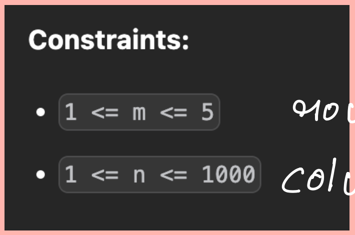
→ This is → exploring all the ways from the options.

- what concept comes to your mind ?

- RECURSION Possibly improve by memoization.

OKAY !

Now, the challenge is, How do you implement /approach this ?



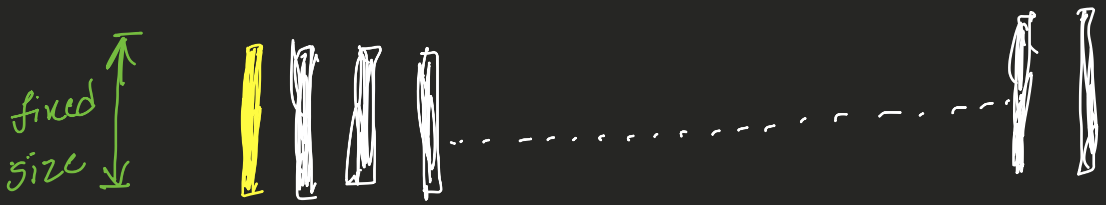
Constraints:

- `1 <= m <= 5` rows
- `1 <= n <= 1000` columns

→ If we see the constraints;

The rows are Limited to 5 !!

So, one possible way to paint the whole grid is:



- Forget about all the rows.

- Just imagine, you have 5 alphabets

[a, b, c, d, e]

Now ask your self - How can make different strings of length 5 so that no adjacent alphabets follow the same sequence in real alphabet sequence.

Basically, you should not see something Like (a, b...

or (...bc...) or (... de)

- This is Basic 1-D recursion right.

This is where the intuition comes from.

Story points flow Summary:

- find all the Combination of {R, G, B} colors so that the particular column satisfies the criteria.

eg: For a length 3

$$\left[\begin{array}{c}R\\G\\B\end{array}, \begin{array}{c}R\\B\\G\end{array}, \begin{array}{c}G\\B\\R\end{array}, \begin{array}{c}G\\R\\B\end{array}, \begin{array}{c}B\\G\\R\end{array}, \begin{array}{c}B\\R\\G\end{array}\right]$$

[Now, you try to arrange them in adjacent colums so that no two adjacent colors are same in a row].

- Finally count the ways and return them.
- optimize it by using Memoization through DP.

- Similar problem: Leetcode - 1411

Hopefully, this gives you an idea to gather your thoughts !!