FLOODFILL Solution:

Không mất tổng quát ta có thể coi màu của hai hình vuông liên tiếp là khác nhau bằng dồn các hình vuông liên tiếp có cùng một màu thành một hình vuông. Trong C++ sử dụng lệnh:
n=unique(c+1,c+n+1)-c-1;

Đặt $f[L,R,x]$ là số thao tác ít nhất để biến đổi đoạn màu $c_L,\dots,c_R$ thành màu x. Khi đó:
+) Nếu $x \neq c_L$ và $x \neq c_R$ thì $f[L,R,x] = 2 + f[L+1,R-1,x]$
+) Trường hợp còn lại $f[L,R,x] \leq 1 + f[L+1,R-1,x]$
Do vậy phương án tối ưu (biến đổi với số lần ít nhất) là phương án xảy ra khi $x = c_L$ hoặc $x = c_R$.
Đặt $dp[L,R,1]$ là số thao tác ít nhất biến đổi đoạn $c_L,\dots,c_R$ thành màu c_L
Đặt $dp[L,R,2]$ là số thao tác ít nhất để biến đổi đoạn $c_L,\dots,c_R$ thành màu c_R
Ta có công thức:

$$dp[L,R,1] = \min\{dp[L+1,R,1] + (c_L \neq c_{L+1}), dp[L+1,R,2] + (c_L \neq c_R)\}$$
$$dp[L,R,2] = \min(dp[L,R-1,1] + (c_L \neq c_R), dp[L,R-1,2] + (c_{R-1} \neq c_R)\}$$

Chú ý kỹ thuật code (tránh đệ qui có nhớ):

```
for(L=n;L≥1;--L) {
    dp[L,L,1]=dp[L,L,2]=0;
    for(R=L+1;R≤n;++R) {
        ......
    }
}
```