

Chỉ có 2 ma trận $N \times M$ thỏa mãn hai ô chung cạnh khác màu nhau và 2 ma trận đó có dạng: $a1[i][j] = (i + j) \bmod 2$, và $a2[i][j] = 1 - (i+j) \bmod 2$

Ví dụ: với $n = 3$, $m = 5$ chỉ có 2 ma trận là:

01010

10101 $\Rightarrow a1[i][j] = (i + j) \bmod 2$

01010

10101

01010 $\Rightarrow a2[i][j] = 1 - (i + j) \bmod 2$

10101

Gọi ma trận được nhập vào là a

Ta sẽ duy trì 2 biến:

$r1$ là số vị trí (i,j) mà $a[i][j]$ khác $(i + j) \bmod 2$ ($a[i][j]$ khác $a1[i][j]$)

$r2$ là số vị trí (i,j) mà $a[i][j]$ khác $1 - (i + j) \bmod 2$ ($a[i][j]$ khác $a2[i][j]$)

Kết quả sẽ là $\min(r1, r2)$

- Đối với sub2, ta sẽ trải mảng 2 chiều $a[n][m]$ thành mảng 1 chiều $a[n*m]$ bằng cách đổi vị trí (i,j) thành $(i - 1)*m + j$. Ví dụ: với $n = 2$, $m = 3$, ta sẽ sử dụng $a[5]$ thay cho $a[2][2]$

- Đối với sub3, ta không cần sử dụng mảng mà chỉ sử dụng một biến char để nhập dữ liệu.

-Lưu ý: trong C++, có thể sử dụng phép $\&1$ để tính số dư khi chia cho 2. Ví dụ: $5 \& 1 = 1$

Tìm hiểu thêm về các phép xử lý bit: <https://vnoi.info/wiki/translate/topcoder/fun-with-bits>