# November 2011 Long Contest on Codechef
# Problem DOMNOCUT editorial

Clearly the product $mn$ should be even. Otherwise the answer is IMPOSSIBLE. We start with the following lemma that provides the lower bound for the minimal number of cuts for any grid.

**Lemma.** Any domino tiling of the $n \times m$ grid has at least

- $(m + n - 2 - mn/4)_+$ cuts if $n, m$ are even;
- $(m/2 + n - 2 - m(n-1)/4)_+$ cuts if $n$ is odd and $m$ is even,

where $x_+ = \max\{x, 0\}$.

**Proof.** Consider some vertical or horizontal line $l$ that passes through the grid. It divides the grid into two parts. We call the line odd if these parts have odd areas and even otherwise. Let $N_l$ be the number of dominos that this line cuts. It is easy to see that $N_l$ has the same parity as the line. Note that for each domino there is exactly one line that cuts it. Hence the sum of $N_l$ for all lines is equal to the total number of domino, that it $\sum_l N_l = mn/2$. Let $K_1$ be the number of odd lines and $K_2$ be the number of even lines. Let $C$ be the number of cuts, that is, the number of lines that doesn't cut any domino. Clearly all these line are even. Hence $mn/2 = \sum_l N_l \geq K_1 + 2(K_2 - C) = K_1 + 2K_2 - 2C$ (this is because each of the $K_1$ odd lines cuts at least one domino and each of the remaining $K_2 - C$ even lines cuts at least 2 dominos). So

$$C \geq K_1/2 + K_2 - mn/4 \qquad (*)$$

Now it is left to count $K_1$ and $K_2$ for each grid. Note that $K_1 + K_2 = m + n - 2$ (the total number of lines).

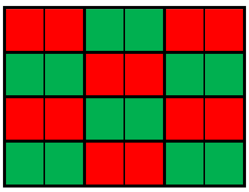
If $m$ and $n$ are even then it is easy to see that all lines are even. Hence $K_1 = 0$ and $K_2 = m + n - 2$. And we obtain the desired estimate on $C$ in this case.

If $n$ is odd and $m$ is even then line is odd if and only if it cuts the grid into grids of sizes $n \times k$, $n \times (m-k)$ where $k$ is odd. Hence $K_1 = m/2$ and $K_2 = m/2 + n - 2$. Substituting this in (*) we obtain the required estimate in this case also. Thus Lemma is proved. ■

It is interesting to note that these estimates are sharp. That is for each $m, n$ there exists a domino tiling of $n \times m$ grid with mentioned in the Lemma number of cuts.

Note also that we can assume that $n \leq m$.

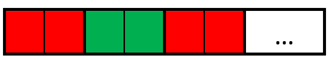
Now let's find the minimal number of colors. Clearly one color will be enough only for the grid $1 \times 2$. Otherwise we need at least two colors. Further note that domino tiling can be colored in two colors only if it has the form



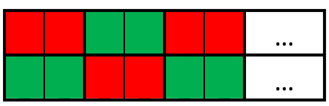
It is easy to see that this tiling will have the minimal number of cuts only in the cases $1 \times 2k$ and $2 \times 2k$. Hence for all other boards we need at least three colors. And as we see later it always suffices to have 3 colors.

Now let's find for each grid the corresponding minimal tiling. In what follows there will be only pictures. You can easily verify for each grid that the number of cuts is exactly as in Lemma and number of colors is 3 besides the first two obvious cases.

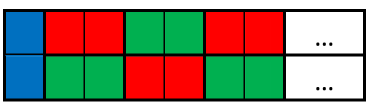
$n = 1, m = 2k$ ($k - 1$ cuts)



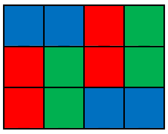
$n = 2, m = 2k$ ($k$ cuts)



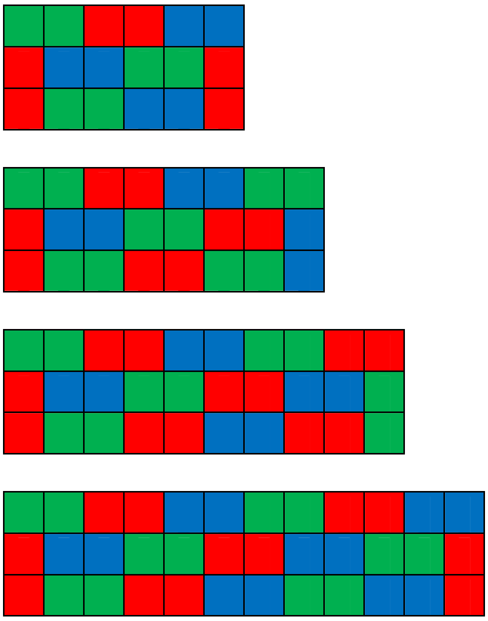
$n = 2, m = 2k + 1$ ($k$ cuts)



$n=3, m=4$ (1 cut)

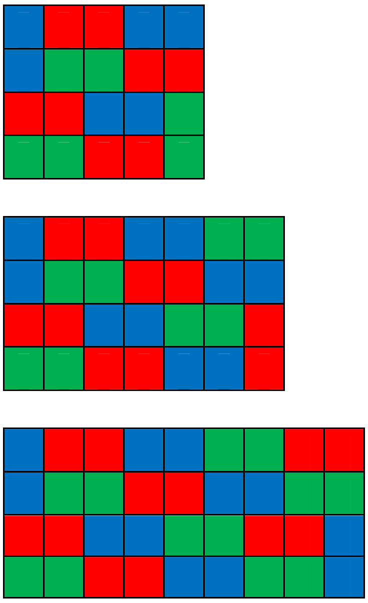


$n=3, m=2k, k \geq 3$ (1 cut)



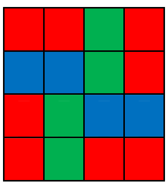
........................................

$n=4, m=2k+1, k \geq 2$ (1 cut)

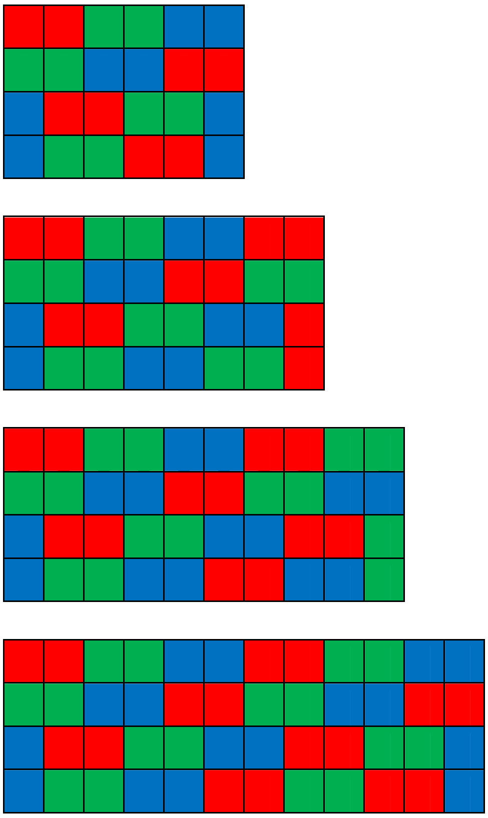


............................

$n=m=4$ (2 cuts)

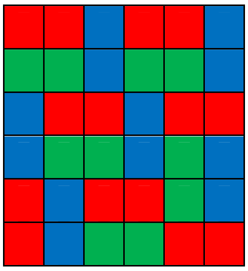


$n = 4, m = 2k, k \geq 3$ (2 cuts)



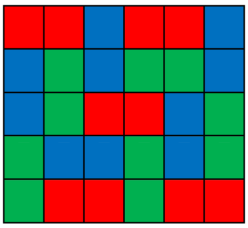
……………………………………

$n = m = 6$ (1 cut)

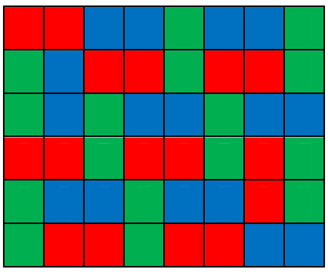


In all other cases there exist tilings with no cuts.

$n = 5, m = 6$ (no cuts)

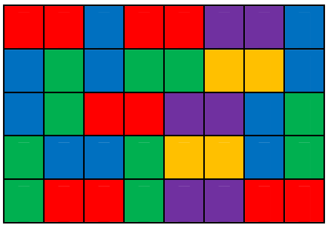


$n = 6, m = 8$ (no cuts)



I would say that I couldn’t find the last three tilings by pencil and paper: all tilings with the minimal number of cuts that I’ve created can’t be colored by 3 colors. So I even believed for some period of time that the minimum number of colors is 4. But then I wrote a bitmask dp to generate all tilings with minimal number of cuts and try to color them in three colors by bruteforce and finally found these three tilings as well as three-colored tilings with no cuts for other small grids.

But how to handle larger grids? The idea is the following. If we have tiling with no cuts for $n \times m$ grid then we can create a tiling with no cuts for $n \times (m+2)$ grid in a very simple way. See for example how it is made for transformation from $5 \times 6$ grid to $5 \times 8$ grid:



In this way using tilings for basic grids $5 \times 6$ and $6 \times 8$ we can obtain tiling for any other grid with size $n \times m$ where $n \geq 5$ and $m \geq 6$ (except the board $6 \times 6$ of course) by the sequence of such operations.

But now the problem is to obtain a 3-coloring of this new tiling. As you see from the picture we can't simply use red, green and blue instead of violet and yellow. We also need to change colors of old dominos in last two columns.

Here three scenarios are possible.

First one is implemented in author solution. You can hard-code this general tiling in both cases ($n, m$ are even and $n, m$ are of different parity) and then use backtracking for its 3-coloring. This tiling is specific and simple backtracking find 3-coloring very quickly. However there are some pitfalls. You should use DFS to color all that possible when you pick the color for some domino. You should save the changes that you made in array where colors are stored and before exit in one recursive step undo these changes. I should warn you that copying of the whole array of colors is very slow and requires a lot of memory.

Second scenario is to restore the coloring at each step. This approach is implemented in Tiancheng Lou's (aka ACRush) solution. BTW, I recommend this solution as the best and the simplest solution for this problem.

Finally you can try to find the pattern for colors manually. Many contestants have used this approach. See for example a "magic" solution of Kazuhiro Hosaka (aka lyrically). To find the pattern he uses some "magic" arrays of constants magic2[], magic6[] and magic8[] ☺.