J213 Paint the Wall

Author: Alex Tung
Setter: Hei Chit Cheng
Solution: Ian Wong

Problem

Given a R*C grid, initially all cells are white, paint K cells to black. Maximize pairs of adjacent cells with different color.

```
3 3 3

010
100
010
```

For all cases:

$1 \leq R, C \leq 100$

$0 \leq N \leq R \times C$

	Points	Constraints
1	13	$R = 1$
2	18	$R = 2$ $2 \leq C \leq 100$
3	9	$R = C = 3$
4	8	$R \times C$ is even $N = \frac{R \times C}{2}$
5	29	$R \times C$ is even
6	23	No additional constraints

Stats

First solve: cwong 1:14

7 contestants had scored 100

Mean: 22

Subtask 3

R = C = 3

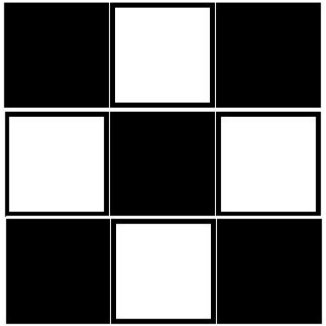
You can solve K=1..9 on paper and hardcode it, or writing a brute force algorithm to generate all possible colourings and find the optimal one.

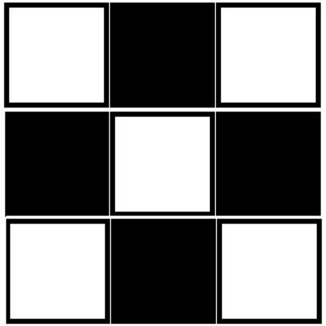
Time complexity: O(1) or O(2^(R * C))

Main Observation 1

Notice that when K > R * C / 2, the problem can be transformed to, initially all cells are ‘1’, we are changing R * C - K cells to ‘0’. So, we can solve the original problem with K = R * C - K and flip the cell color at last.

From now on, we assume that K <= R * C / 2.





Subtask 1

R = 1

Intuitively, we know when C is odd, we should choose cell 2, 4, ..., C - 1. (row 1)

And when C is even, we can choose 1, 3, ..., C - 1 or 2, 4, ..., C.

Why the parity of C matters? Think about C = 3, we have to choose the middle cell since only it has two neighbours. In general, we don't really want to choose cell 1 or C unless we have no choice.

Main Observation 2

When K <= R * C / 2, in optimal answer, we will never paint two adjacent cells with ‘1’. We can always construct such answer (choose odd or even columns).

Subtask 1

Since in both cases (C is even or odd), start choosing from 2 is optimal.

So we will paint cell 2, 4, 6, ... and stop when we have painted K cells.

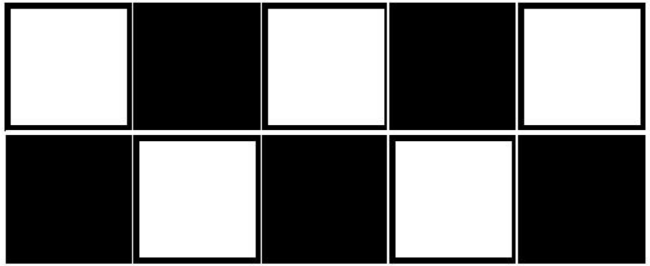
Time complexity: O(C)

Subtask 2

R = 2, C <= 100

We can extend main observation 2, and it also works when R = 2.

When K = R * C / 2, we know our answer will be:



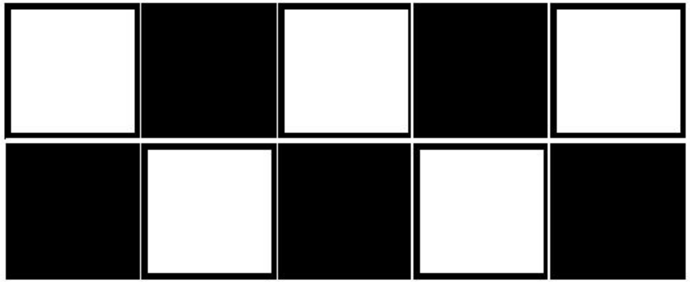
What if K < R * C / 2 ?

Main Observation 3

We can pick the cells greedily.

When R = 2 and C = 5, we have 5 choices to paint.

Notice that we can consider these choices independently and it wouldn't affect others, as we would never paint both adjacent cells with '1'.

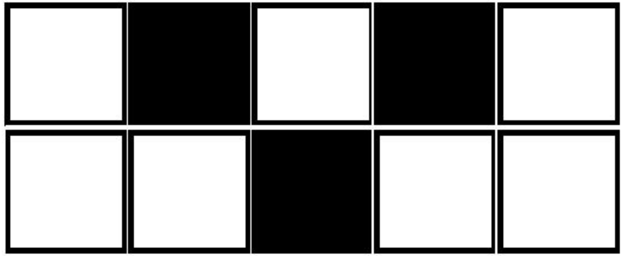


Subtask 2

We want to greedily paint cells that have more adjacent neighbours (cells that aren't located in column 1 or C).

For example, when R = 2, C = 5 and K = 3, the solution below is one of the optimal solutions.

Time complexity: O(R * C)



Subtask 4

R * C is even and K = R * C / 2.

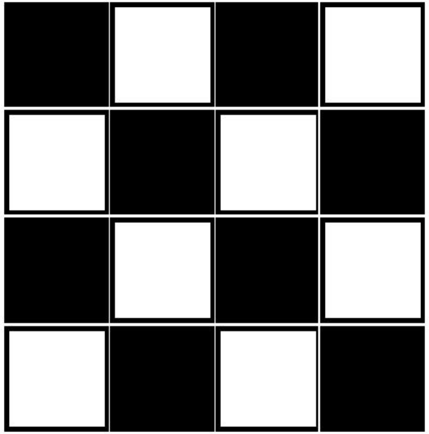
With our intuition or the observation we have, main observation 2 and indeed it works in general case (R > 2), we can notice that we will be painting the grid like this:

(1, 1), (1, 3), ...

(2, 2), (2, 4), ...

(3, 1), (3, 3), ...

...



Time Complexity: O(R * C)

Subtask 5

R * C is even

We can combine our idea in subtask 2 and 4. We are picking K non-adjacent cells to paint them as ‘1’ and we are picking them greedily by their number of adjacent neighbours. So we are picking K cells from here:

(1, 1), (1, 3), …

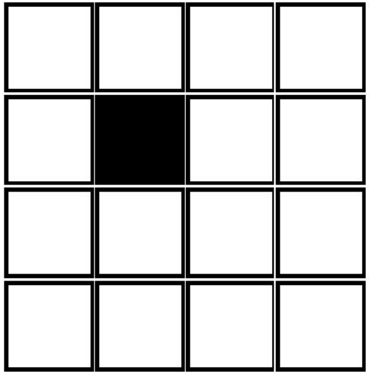
(2, 2), (2, 4), …

(3, 1), (3, 3), …

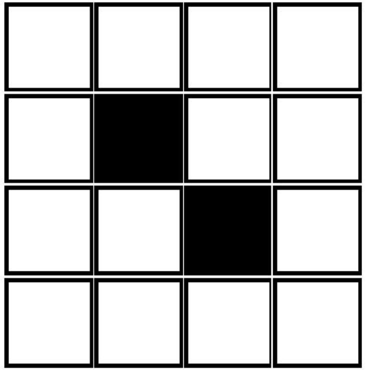
…

Subtask 5

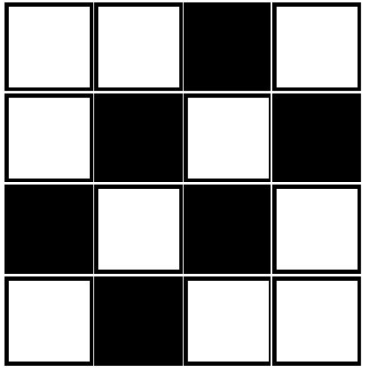
K = 1



K = 2



K = 6



Time complexity: O(R * C)

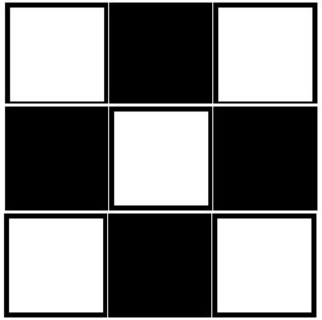
Subtask 6

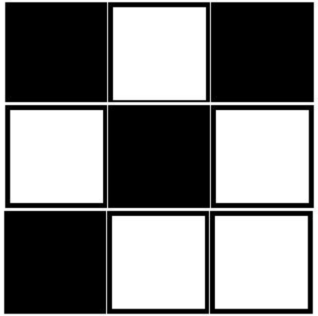
No additional constraints

Why doesn’t subtask 5’s idea work in general?

When R = C = 3, K = 4:

it is better to paint it in the way of the left one than the right one.





Main Observation 4

We have two (and only two) different choosing mechanisms for non-adjacent cells:

1. choose cell (i, j) where i + j = 0 (mod 2)
2. choose cell (i, j) where i + j = 1 (mod 2)

When R * C is even, two methods are the same. (Imagine R = 4 and C = 4)

When R * C is odd, one might yield a better result.

Subtask 6

We try both methods and pick the one with larger result.

Time complexity: O(R * C)