BACK

# Chess Bishop Dream

180,554

DESCRIPTION MY SOLUTIONS LEADERBOARD COMMENTS README

CODEWRITING SCORE: 300/300

In *ChessLand* there is a small but proud chess bishop with a recurring dream. In the dream the bishop finds itself on an `n × m` chessboard with mirrors along each edge, and it is not a bishop but a ray of light. This ray of light moves only along diagonals (the bishop can't imagine any other types of moves even in its dreams), it never stops, and once it reaches an edge or a corner of the chessboard it reflects from it and moves on.

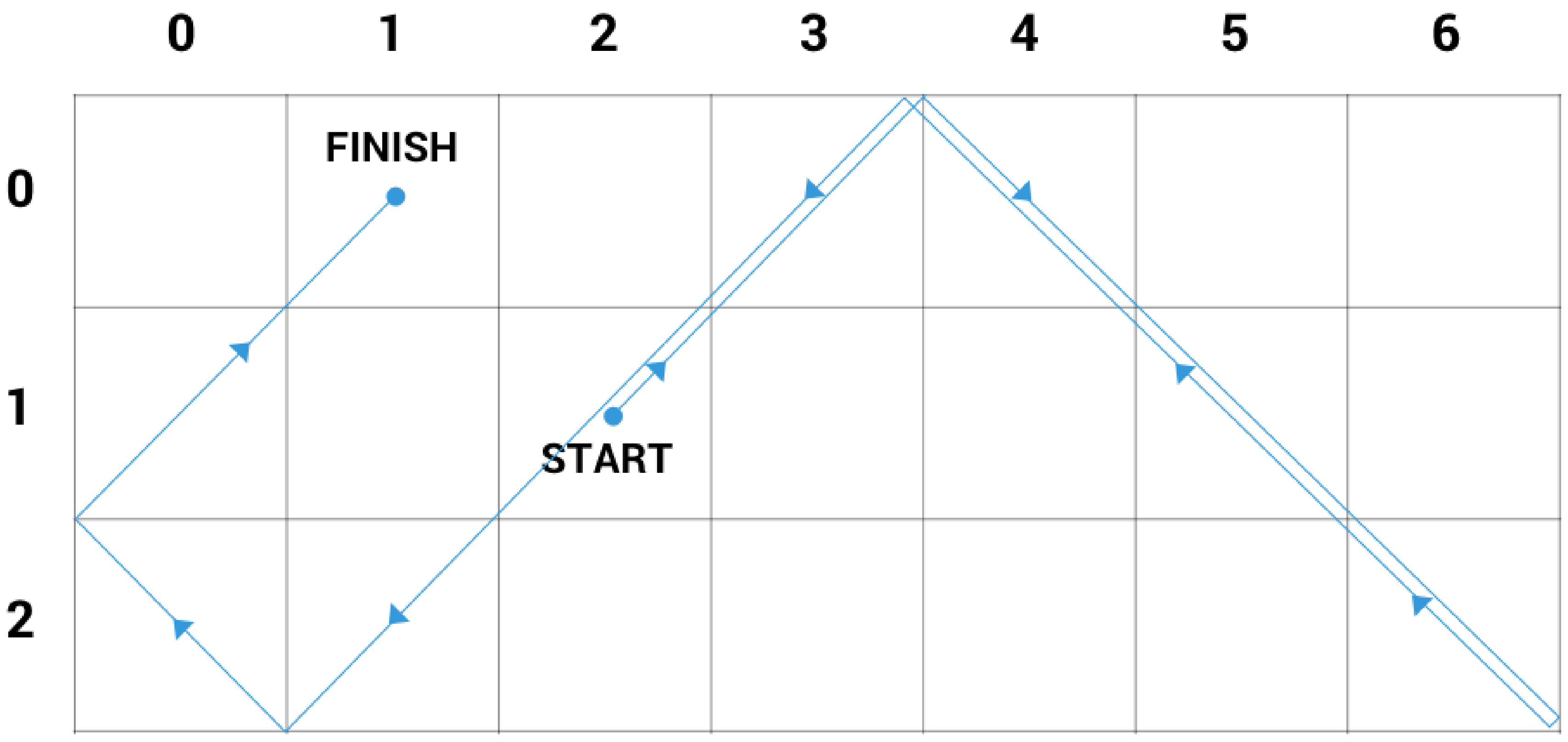
Given the initial position and the direction of the ray, find its position after `k` steps where a step means either moving from one cell to the neighboring one or reflecting from a corner of the board.

**Example**

For `boardSize = [3, 7]`, `initPosition = [1, 2]`, `initDirection = [-1, 1]` and `k = 13`, the output should be `chessBishopDream(boardSize, initPosition, initDirection, k) = [0, 1]`.

Here is the bishop's path:

```
[1, 2] -> [0, 3] -(reflection from the top edge)-> [0, 4] ->
[1, 5] -> [2, 6] -(reflection from the bottom right corner)-> [2, 6] ->
[1, 5] -> [0, 4] -(reflection from the top edge)-> [0, 3] ->
[1, 2] -> [2, 1] -(reflection from the bottom edge)-> [2, 0] -(reflection from the left edge)->
[1, 0] -> [0, 1]
```



**Input/Output**

- **[execution time limit] 4 seconds (js)**
- **[input] array.integer boardSize**

  An array of two integers, the number of rows and columns, respectively. Rows are numbered by integers from `0` to `boardSize[0] - 1`, columns are numbered by integers from `0` to `boardSize[1] - 1` (both inclusive).

  *Guaranteed constraints:*
  `1 ≤ boardSize[i] ≤ 20`.

- **[input] array.integer initPosition**

  An array of two integers, indices of the row and the column where the bishop initially stands, respectively.

  *Guaranteed constraints:*