**Snakes and Ladders:**

On an N x N board, the numbers from 1 to N\*N are written *boustrophedonically* **starting from the bottom left of the board**, and alternating direction each row.  For example, for a 6 x 6 board, the numbers are written as follows:



You start on square 1 of the board (which is always in the last row and first column).  Each move, starting from square x, consists of the following:

* You choose a destination square S with number x+1, x+2, x+3, x+4, x+5, or x+6, provided this number is <= N\*N.
  + (This choice simulates the result of a standard 6-sided die roll: ie., there are always **at most 6 destinations, regardless of the size of the board**.)
* If S has a snake or ladder, you move to the destination of that snake or ladder.  Otherwise, you move to S.

A board square on row r and column c has a "snake or ladder" if board[r][c] != -1.  The destination of that snake or ladder is board[r][c].

Note that you only take a snake or ladder at most once per move: if the destination to a snake or ladder is the start of another snake or ladder, you do **not** continue moving.  (For example, if the board is `[[4,-1],[-1,3]]`, and on the first move your destination square is `2`, then you finish your first move at `3`, because you do **not** continue moving to `4`.)

Return the least number of moves required to reach square N\*N.  If it is not possible, return -1.

**Example 1:**

**Input:** [

[-1,-1,-1,-1,-1,-1],

[-1,-1,-1,-1,-1,-1],

[-1,-1,-1,-1,-1,-1],

[-1,35,-1,-1,13,-1],

[-1,-1,-1,-1,-1,-1],

[-1,15,-1,-1,-1,-1]]

**Output:** 4

**Explanation:**

At the beginning, you start at square 1 [at row 5, column 0].

You decide to move to square 2, and must take the ladder to square 15.

You then decide to move to square 17 (row 3, column 5), and must take the snake to square 13.

You then decide to move to square 14, and must take the ladder to square 35.

You then decide to move to square 36, ending the game.

It can be shown that you need at least 4 moves to reach the N\*N-th square, so the answer is 4.

**Note:**

1. 2 <= board.length = board[0].length <= 20
2. board[i][j] is between 1 and N\*N or is equal to -1.
3. The board square with number 1 has no snake or ladder.
4. The board square with number N\*N has no snake or ladder.