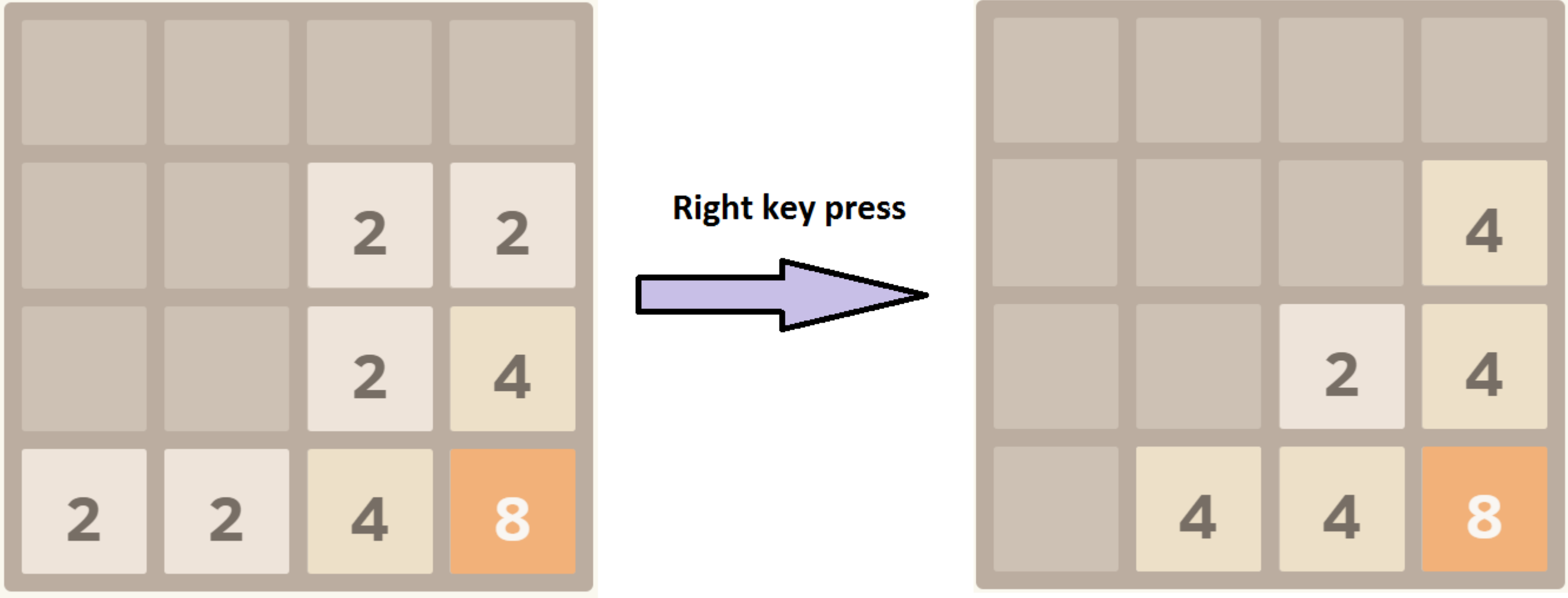


You are most likely familiar with the game 2048 .

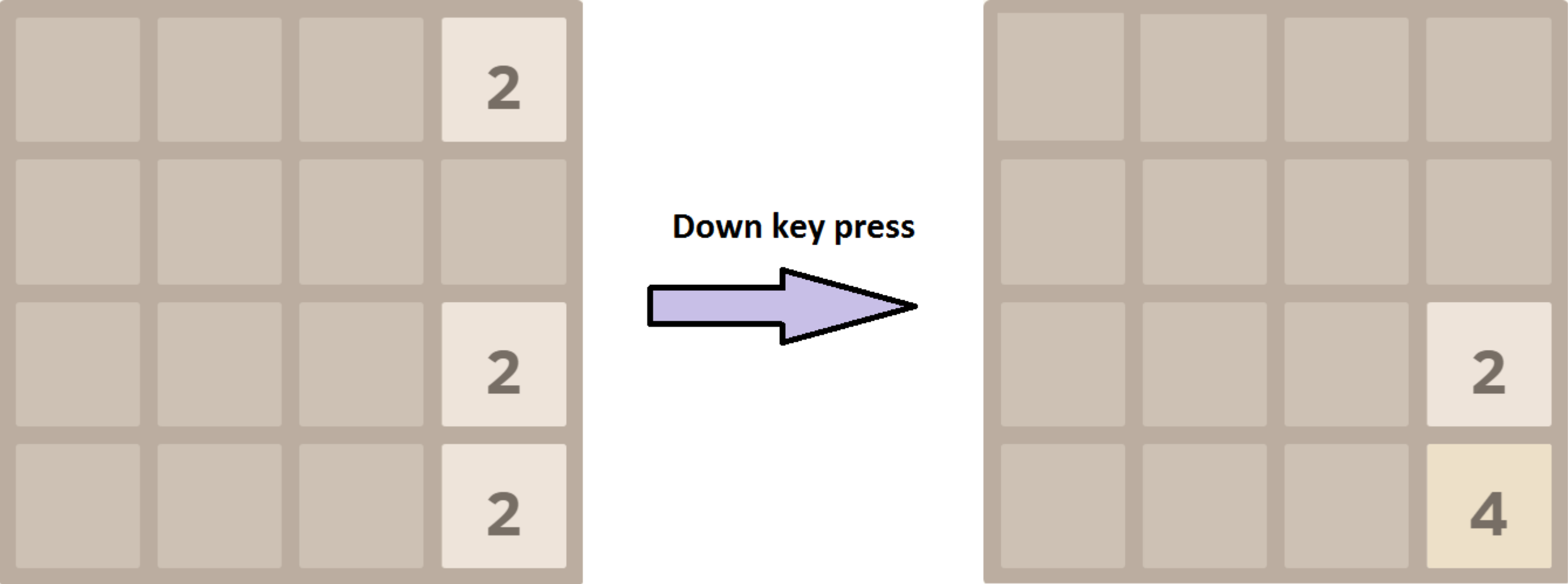
2048 is played on a gray 4 × 4 grid with numbered tiles that slide smoothly when a player moves them using one of the four arrow keys - Up , Down , Left or Right . On every turn, a new tile with a value of either 2 or 4 randomly appears on an empty spot of the board. After one of the keys is pressed, the tiles slide as far as possible in the chosen direction until they are stopped either by another tile or by the edge of the grid. If two tiles with the same number collide while moving, they merge into a tile with this number doubled. You can't merge an already merged tile in the same turn. If there are more than 2 tiles in the same row (column) that can merge, the farthest (closest to an edge) pair will be merged first (see the second example).

In this problem you are not going to solve the 2048 puzzle, but you are going to find the final state of the board from the given one after a defined set of n arrow key presses, assuming that no new random tiles will appear on the empty spots.

The following example shows the next state of the board after pressing Right .



This example shows the next state of the board after pressing Down .



For more details you can visit <http://gabrielecirulli.github.io/2048/> and play online :)

You are given a matrix 4 × 4 which corresponds to the 2048 game grid. grid[0][0] corresponds to the upper left tile of the grid. Each element of the grid is equal to some power of 2 if there is a tile with that value in the corresponding position, and 0 if it corresponds to the empty spot. You are also given a sequence of key presses as a string path . Each character of the string equals L , R , U , or D corresponding to Left , Right , Up or Down respectively. Please note that in some cases after pressing an arrow key nothing will be changed.

Example

For

```
grid = [[0, 0, 0, 0],
        [0, 0, 2, 2],
        [0, 0, 2, 4],
        [2, 2, 4, 8]]
```

and path = "RR" , the output should be

```
game2048(grid, path) = [[0, 0, 0, 0],
                        [0, 0, 0, 4],
                        [0, 0, 2, 4],
                        [0, 0, 8, 8]]
```

Input/Output

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

A square matrix of size 4 × 4 , the starting state of the board. Each value of the matrix is the power of 2 .