



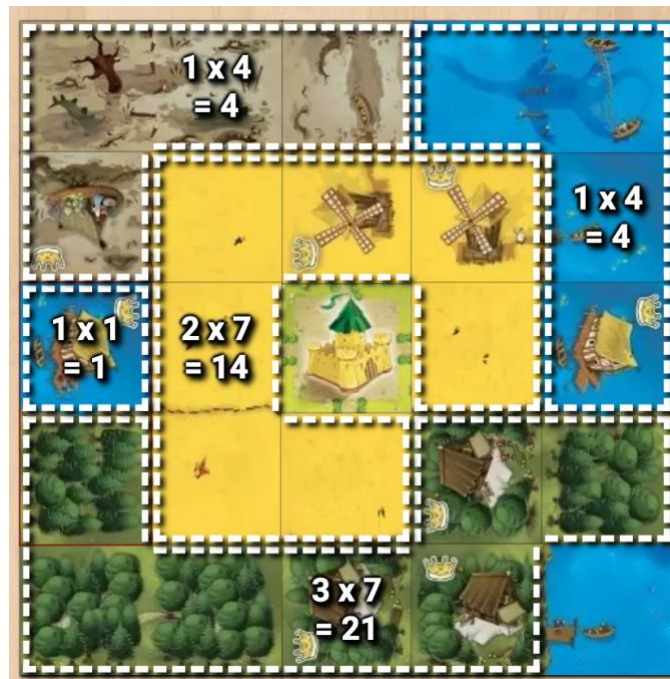
Zanbato

Trust in Private Markets

Introduction

King Domino is a board game where you try to lay down dominos representing various terrain types to score the most points. Specifically in King Domino, you get points for having contiguous areas with crowns on them.

Your job is to take a completed board and figure out how many points it scores. As an example, here's a completed board



For each contiguous area (as shown above in the dotted white lines), you get points equal to the number of squares times the number of crowns in that area. For example, the green forests in the bottom has 3 crowns and 7 squares, so it's worth 21 points. The 3 lakes are in separate regions, so they are worth 1, 4, and 0 points. Doing the same across all of the different areas would add up to 44 points for this board.

The input will come as 2 5x5 arrays, one for the terrains and one for the crowns. For the image above, the terrains will appear as

```
1 SSSLL
2 SWWWL
3 LWKWL
4 FWFFF
5 FFFFL
```

Where the possible terrains are:

- s = Swamp
- L = Lake
- W = Wheat
- F = Forest
- G = Grass (does not appear above)
- M = Mine (does not appear above)
- K = Kingdom

And the crowns for the board above are

```
1 00000
2 10110
3 10001
4 00010
5 00110
```

Note that up to 3 crowns can appear on a single tile.

Given these 2 arguments, return an integer for the number of points the board will score.

Edge Cases

- Contiguous areas are defined as touching on one of the 4 sides. If two squares only touch at a corner, then they are not considered part of the same area
- In actual King Domino, it's possible for the board to not be a 5x5 or for there to be gaps. Do not worry about those cases. Assume that it will be exactly 5x5 and all of the squares are filled
- The Kingdom square is typically, but does not have to be, in the middle. However, it's not worth any points in any case

What We're Looking For

1. Provide your code along with any test cases you come up with as part of your solution
2. Although the problem itself is designed to solve a complete 5x5 grid, we expect the solution to generalize to larger grids e.g. 7x7
3. Given that, we believe the optimal solution is $O(n^2)$ where n is the width (or length) of the grid
4. We expect this problem to take about one hour to complete
5. Although the input is supposed to come as arrays, you may find it useful to use strings to represent your test data and then a helper to parse that into the desired array format
6. Since we will be reading your code, please make sure your code is generally readable. You do not necessarily need to include formatted docs, comments, linting, etc. but use meaningful variable names and decompose functions as you would find reasonable