Tile

Solution

This problem can be solved by a dynamic programming. We define a function $s(x, y)$ to be the size of the largest square with (i, j) as its lower-left corner. There are two cases in the recursion of s.

- If $s(x+1, y) = s(x, y+1) = m$ then $s(x, y)$ will be $m+1$ if the grid at $(x+m, y+m)$ is usable, and $s(x, y)$ will be m if the grid at $(x+m, y+m)$ is defective.
- If $s(x+1, y) \neq s(x, y+1)$, then it is easy to see that $s(x, y)$ is $\min(s(x+1, y), s(x, y+1)) + 1$.
- If either x or y is $n-1$, i.e., it is at the top row or right column, then $s(x, y)$ is 1 if the grid is usable, or 0 otherwise.

It is easy to see that the dynamic programming runs in $O(n^2)$ times, where n is the size of the materials. Then we can scan s for the largest value and count them to find the answer.