750. Number Of Corner Rectangles

DescriptionHintsSubmissionsDiscussSolution

Given a grid where each entry is only 0 or 1, find the number of corner rectangles.

A *corner rectangle* is 4 distinct 1s on the grid that form an axis-aligned rectangle. Note that only the corners need to have the value 1. Also, all four 1s used must be distinct.

Example 1:

```
Input: grid =
[[1, 0, 0, 1, 0],
 [0, 0, 1, 0, 1],
 [0, 0, 0, 1, 0],
 [1, 0, 1, 0, 1]]
Output: 1
Explanation: There is only one corner rectangle, with corners grid[1][2], grid[1][4], grid[3][2], grid[3][4].
```

Example 2:

```
Input: grid =
[[1, 1, 1],
 [1, 1, 1],
 [1, 1, 1]]
Output: 9
Explanation: There are four 2x2 rectangles, four 2x3 and 3x2 rectangles, and one 3x3 rectangle.
```

Example 3:

```
Input: grid =
[[1, 1, 1, 1]]
Output: 0
Explanation: Rectangles must have four distinct corners.
```

Note:

1. The number of rows and columns of `grid` will each be in the range `[1, 200]`.
2. Each `grid[i][j]` will be either `0` or `1`.
3. The number of `1`s in the grid will be at most `6000`.

```
int countStart(vector<vector<int>>& grid, int x, int y, int rows, int cols)
{
      int count=0;
      for(int j=y+1;j<cols;j++)
      {
            if(grid[x][j]==1)
            {
                  for(int i=x+1;i<rows;++i)
                  {
                        if(grid[i][j]==1 && grid[i][y]==1) count++;
                  }
            }
```

```
	}
	return count;
}

int countCornerRectangles(vector<vector<int>>& grid) {
	int n = grid.size();
	int m = grid[0].size();
	int count = 0;
	for(int i=0;i<n;++i)
	{
		for(int j=0;j<m;++j)
		{
			if(grid[i][j]==1)
			{
				count+=countStart(grid,i,j,n,m);
			}
		}
	}
	return count;
}
```