Kick Start 2021 - Round A

Analysis: L Shaped Plots

Test set 1

In order to verify that a segment is good, we need to check whether all the cells in that segment contain 1 or not. To check whether all the cells in a segment contain 1 or not, we can calculate prefix sum of the matrix and then check whether sum of cells on this segment is equal to the length of the segment or not. Let $query(a, b, c, d)$ denote the sum of cells in submatrix with (a, b) as top left corner and (c, d) as bottom right corner. We can calculate this sum in $O(1)$ using the prefix sum matrix. For more details on using prefix sum to calculate sum of cells on a submatrix, please refer this. In order to get the sum of cells on segment from (i, j) to (i, l), we can simply check if $query(i, j, i, l) = |j - l| + 1$.

L-shape comprises of two segments which meet at a common point. Except the common point, consider the other end point of one segment as (i, j) and the end point of the other segment as (k, l). Now these end points could meet at either (i, l) or (k, j) to form an L-shape. So, if we know the end points of each segment, we can figure out the common point where segments would meet.

For each pair of end points (i, j) and (k, l) of segments of L-shape, we already saw that there could be 2 possible meeting points.

- For segments meeting at (i, l), check that $query(i, j, i, l) = |j - l| + 1$ and $query(k, l, i, l) = |i - k| + 1$. Besides, either $|j - l| + 1 = 2 \times (|i - k| + 1)$ or $|i - k| + 1 = 2 \times (|j - l| + 1)$ should be true. If these conditions are satisfied, increase the answer by 1.
- For segments meeting at (k, j), check that $query(i, j, k, j) = |i - k| + 1$ and $query(k, l, k, j) = |j - l| + 1$. Besides, either $|i - k| + 1 = 2 \times (|j - l| + 1)$ or $|j - l| + 1 = 2 \times (|i - k| + 1)$ should be true. If these conditions are satisfied, increase the answer by 1.

We can iterate over all possible end points of these two segments L-shape because there are $O(\mathbf{R}^2 \times \mathbf{C}^2)$ such possible combinations. We can calculate the number of possible L-shapes with these end points in $O(1)$. Hence, the overall complexity of the solution is $O(\mathbf{R}^2 \times \mathbf{C}^2)$.

Test set 2

We cannot iterate over all possible end points of the segments for this test set as the solution would time out. If for each cell, we can quickly calculate how many L-shapes are such that both of its segments meet at this cell, we can iterate over each cell of the matrix only once and calculate our answer. We can safely ignore those cells that have value 0 as they cannot be part of any L-shape.

Consider a cell (i, j). There can be 4 types of L-shapes that have both of its segments meet this cell.

- Type 1: One of the segments is to top of (i, j), and other segment is to the right of (i, j).
- Type 2: One of the segments is to top of (i, j), and other segment is to the left of (i, j).
- Type 3: One of the segments is to bottom of (i, j), and other segment is to the right of (i, j).
- Type 4: One of the segments is to bottom of (i, j), and other segment is to the left of (i, j).

Let $Count(x, y)$ be the number of L-shapes with both its segments meeting at a particular point of which the length of the segment parallel to one axis is x and the length of the segment parallel to the other axis is y. Number of L-shapes with longer segment as part of the segment with length x are $min(\frac{x}{2}, y) - 1$. Similarly, number of L-shapes with longer segment as part of the segment with length y are $min(\frac{y}{2}, x) - 1$. Hence, $Count(x, y) = min(\frac{x}{2}, y) + min(\frac{y}{2}, x) - 2$.

If we can calculate number of consecutive cells that have value 1 in each side of (i, j), we can calculate number of L-shapes of each type with this cell as the common endpoint of the segments using the $Count()$ function above. Let $top(i, j)$ denote the number of consecutive cells that have value 1 including (i, j) and cells above it. For cells with value 0, $top(i, j) = 0$. Formally for cells with value 1, $top(i, j) = i - k + 1$ where $1 \leq k \leq i$ and k is least possible value such that all cells from (k, j) to (i, j) have value 1. Similarly, we can define $bottom(i, j)$, $left(i, j)$ and $right(i, j)$ denoting maximum number of consecutive cells on bottom, left and right of (i, j) respectively.

We can calculate $top(i, j)$ and $left(i, j)$ by iterating from the starting of the matrix and updating their values. Refer to the code below to calculate these values.

```
for(int i = 1; i <= R; i++) {
  for(int j = 1; j <= C; j++) {
     if (matrix[i][j] == 0) continue;
     top[i][j] = top[i - 1][j] + 1;
     left[i][j] = left[i][j - 1] + 1;
  }
  }
```

Similarly, we can calculate $bottom(i, j)$ and $right(i, j)$ by iterating from the end of the matrix. After knowing these values, we can calculate number of L-shapes of each type for cell (i, j). $Count(top(i, j), right(i, j))$, $Count(top(i, j), left(i, j))$, $Count(bottom(i, j), right(i, j))$ and $Count(bottom(i, j), left(i, j))$ denote the number of L-shapes of type 1, 2, 3 and 4 respectively. Thus, for each cell we can add $Count(top(i, j), left(i, j)) + Count(top(i, j), right(i, j)) + Count(bottom(i, j), left(i, j)) + Count(bottom(i, j), right(i, j))$ to the answer.

$top(i, j)$, $bottom(i, j)$, $left(i, j)$ and $right(i, j)$ can be calculated in $O(\mathbf{R} \times \mathbf{C})$ time complexity. $Count(i, j)$ can be calculate in $O(\mathbf{R} \times \mathbf{C})$ time complexity. Finally, we need to iterate over each cell of the matrix, and we can update the answer in $O(1)$ for each cell. Thus, the overall complexity of the solution is $O(\mathbf{R} \times \mathbf{C})$.