1277. Count Square Submatrices with All Ones

We use Dynamic Programming to solve this problem. We traverse row-by-row, from left to right and top to bottom. For each cell (i, j), we record its state as **s(i, j), which denotes the side of the largest square that has all 1s in its cells and bottom right corner at matrix[i][j]**.

**Scenario (1)**: If s(i, j)=t, the following must be true.

(a) matrix[i][j] = 1

(b) matrix[i-k][j-k] = 1

(c) s(i, j-1) = t-1

(d) s(i-1, j) = t-1

The following is obvious. If conditions (a) thru (d) are true, s(i, j)=t. In another word, conditions (a) thru (d) is the sufficient and necessary condition of s(i, j)=t.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |

Now let’s consider the scenarios when conditions (a) thru (d) are not true.

**Scenario (2)**: If (a) is not true, which means matrix[i][j] = 0, then s(i, j)=0.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 0 |

**Scenario (3)**: If (b) is not true but (a), (c), (d) are true, then s(i, j)=t-1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | 0 | 1 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |

**Scenario (4)**: If (c) and/or (d) is not true, let’s say s(i, j-1)=m, s(i-1, j)=n, and m<n, then s(i, j)=m+1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |

To summarize, we have the following state transition function.

1. If matrix[i][j]=1

1.1 If s(i-1, j)=s(i, j-1) and both equal to t-1

1.1.1 Scenario (1): If matrix[i-t][j-t]=1, then s(i, j)=t.

1.1.2 Scenario (3): If matrix[i-t][j-t]=0, then s(i, j)=t-1.

1.2 Scenario (4): If s(i-1, j)=m, s(i, j-1)=n m<n, then s(i, j)=m+1

1.3 Similar to 1.2

2. Scenario (2): If matrix[i][j]=0, then s(i, j)=0.

When we traverse the matrix cell by cell, we sum up the number of sqaures that have all 1s and right bottom at matrix[i][j]. This way, we can avoid adding repetive sqaures. When s(i, j)=t, there are t squares that has all 1s in its cells and right bottom at matrix[i][j]. These squares have side equal to 1, 2, 3, … , t.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| 1 | 0 | 0 | 1 | 1 |
| 2 | 1 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 | 1 |
| 4 | 1 | 1 | 1 | 1 |

For convenience, we add an imaginary row and an imaginary column before the 1st row and the 1st column. The imaginary row and an imaginary column consist of 0s.