Max Lake

Submit solution

My submissions
All submissions
Best submissions

Points: 100 (partial)
Time limit: 1.0s
Memory limit: 512M

Allowed languages
C, C++

You are given a two-dimensional grid with n rows and m columns. Each position in the grid contains a non-negative number, indicating how deep the water is at that spot.

A lake is defined as a group of adjacent cells (connected through horizontal or vertical moves only) where each cell has a depth greater than zero. You cannot pass through or include cells with a depth of zero.

The volume of a lake is calculated by summing the depth values of all the connected, non-zero cells that form it.

Your task is to compute the **maximum** volume of a lake among all such lakes in the grid.

You may use the following logic the contruct the graph.

Copy

```
// node (u) corresponds to the cell (u / m, u % m)

for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j++) {
        if (a[i][j]) {
            if (i + 1 < n && a[i + 1][j]) {
                add_edge(i * m + j, (i + 1) * m + j);
            }
            if (i - 1 >= 0 && a[i - 1][j]) {
                add_edge(i * m + j, (i - 1) * m + j);
            }
            if (j + 1 < m && a[i][j + 1]) {
                add_edge(i * m + j, i * m + (j + 1));
            }
            if (j - 1 >= 0 && a[i][j - 1]) {
                add_edge(i * m + j, i * m + (j - 1));
            }
        }
    }
}
```

Input Format:

- The first line contains an integer t, the number of testcases.
- The first line of each testcase contains two integers n and m. The number of rows and columns in the grid.
- Then n lines follow each with m integers a_{ij} denoting the depth of the water at each cell.

Output Format:

- Output a single integer, the largest volume of a lake in the grid (0 if there is no lake).

Constraints

- $1 \leq n, m \leq 10^3$
- $0 \leq a_{ij} \leq 10^3$
- The sum of $n \times m$ over all testcases does not exceed 10^6

Sample Testcase

Input

Copy

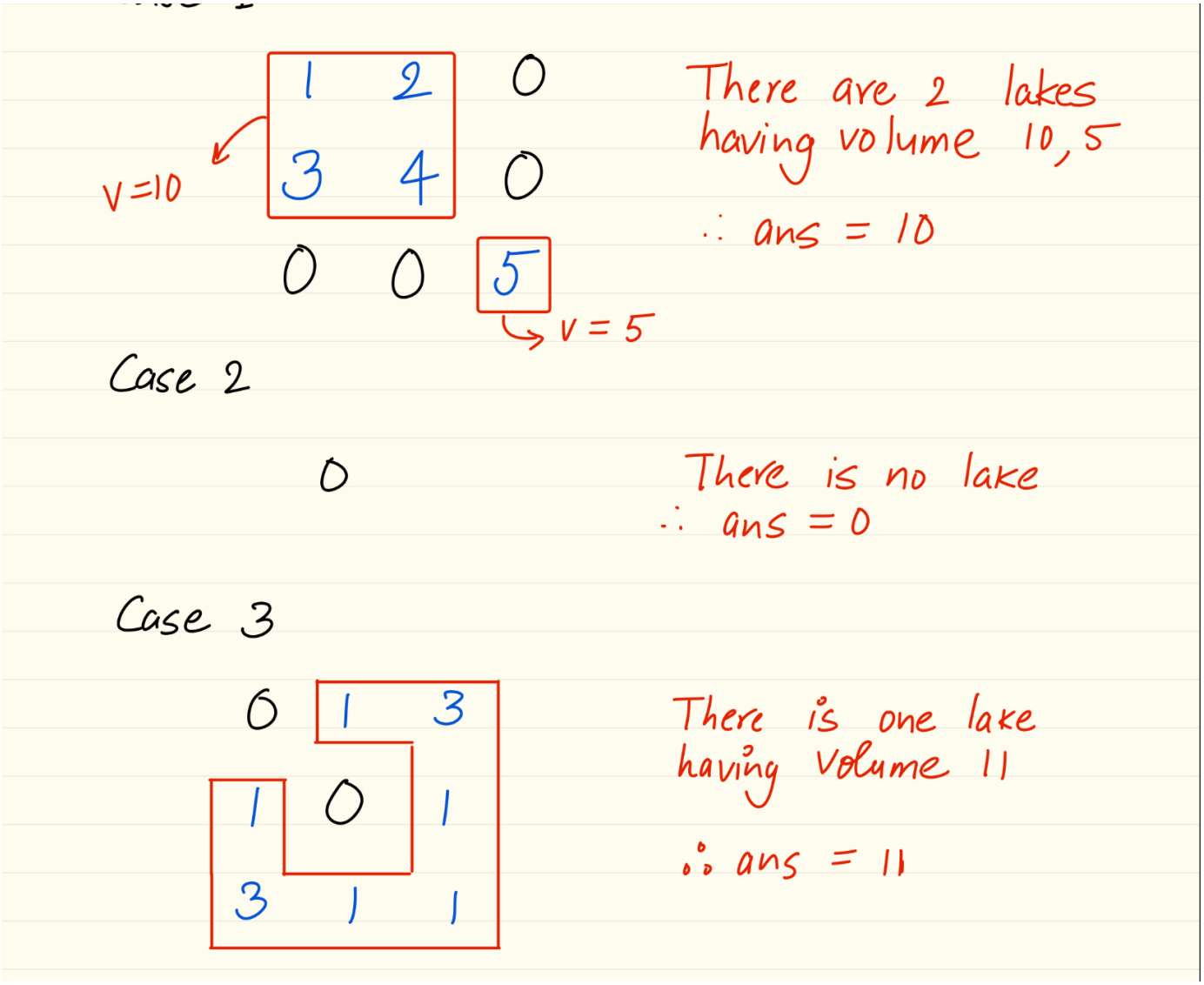
```
3 3
1 2 0
3 4 0
0 0 5
1 1
0
3 3
0 1 3
1 0 1
3 1 1
5 5
1 1 1 1 1
1 0 0 0 1
1 0 5 0 1
1 0 0 0 1
1 1 1 1 1
5 5
1 1 1 1 1
1 0 0 0 1
1 1 1 1 1
1 0 0 0 1
1 1 1 1 1
```

Output

Copy

```
10
0
11
16
19
```

Explanation for Sample Testcase



Clarifications

Request clarification

No clarifications have been made at this time.