# 6111. Spiral Matrix IV

My Submissions (/contest/weekly-contest-300/problems/spiral-matrix-iv/submissions/) Back to Contest (/contest/weekly-contest-300/)

| | |
|---|---|
| User Accepted: | 0 |
| User Tried: | 0 |
| Total Accepted: | 0 |
| Total Submissions: | 0 |
| Difficulty: | Medium |

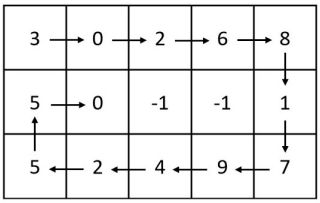
You are given two integers `m` and `n`, which represent the dimensions of a matrix.

You are also given the `head` of a linked list of integers.

Generate an `m x n` matrix that contains the integers in the linked list presented in **spiral** order **(clockwise)**, starting from the **top-left** of the matrix. If there are remaining empty spaces, fill them with `-1`.

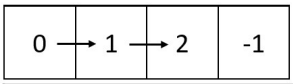
Return *the generated matrix*.

**Example 1:**




```
Input: m = 3, n = 5, head = [3,0,2,6,8,1,7,9,4,2,5,5,0]
Output: [[3,0,2,6,8],[5,0,-1,-1,1],[5,2,4,9,7]]
Explanation: The diagram above shows how the values are printed in the matrix.
Note that the remaining spaces in the matrix are filled with -1.
```

**Example 2:**




```
Input: m = 1, n = 4, head = [0,1,2]
Output: [[0,1,2,-1]]
Explanation: The diagram above shows how the values are printed from left to right in the matrix.
The last space in the matrix is set to -1.
```

**Constraints:**

- `1 <= m, n <= 10^5`
- `1 <= m * n <= 10^5`
- The number of nodes in the list is in the range `[1, m * n]`.
- `0 <= Node.val <= 1000`

JavaScript

```
const getAllData = (list) => {
    let res = [];
    let current = list;
    while (current) {
        res.push(current.val);
        current = current.next;
    }
    return res;
};

const initialize2DArray = (n, m) => { let d = []; for (let i = 0; i < n; i++) { let t = Array(m).fill(-1); d.push(t); }
return d; };

const fillSpiralMatrix = (g, a) => {
    let n = g.length, m = g[0].length, i = 0, j = 0, move = 'r';
    for (const x of a) {
        if (move == 'r') {
            g[i][j] = x;
            if (j + 1 < m && g[i][j + 1] == -1) {
                j++;
```

```
            } else {
                move = 'd';
                i++;
            }
        } else if (move == 'l') {
            g[i][j] = x;
            if (j - 1 >= 0 && g[i][j - 1] == -1) {
                j--;
            } else {
                move = 'u';
                i--;
            }
        } else if (move == 'd') {
            g[i][j] = x;
            if (i + 1 < n && g[i + 1][j] == -1) {
                i++;
            } else {
                move = 'l';
                j--;
            }
        } else if (move == 'u') {
            g[i][j] = x;
            if (i - 1 >= 0 && g[i - 1][j] == -1) {
                i--;
            } else {
                move = 'r';
                j++;
            }
        }
    }
};

const spiralMatrix = (n, m, head) => {
    let a = getAllData(head), g = initialize2DArray(n, m);
    fillSpiralMatrix(g,a)
    return g;
};
```

Custom Testcase | Use Example Testcases

Run | Submit

Submission Result: Accepted (/submissions/detail/737063381/) | More Details > (/submissions/detail/737063381/)

Share your acceptance!