Problem 2326: Spiral Matrix IV

Problem Information

Difficulty: Medium
Acceptance Rate: 82.25%
Paid Only: No
Tags: Array, Linked List, Matrix, Simulation

Problem Description

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 <= 105` * `1 <= m * n <= 105` * The number of nodes in the list is in the range `[1, m * n]`. * `0 <= Node.val <= 1000`

Code Snippets

C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
vector<vector<int>> spiralMatrix(int m, int n, ListNode* head) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public int[][] spiralMatrix(int m, int n, ListNode head) {
```

```
}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def spiralMatrix(self, m: int, n: int, head: Optional[ListNode]) ->
List[List[int]]:
```