Problem 2643: Row With Maximum Ones

Problem Information

Difficulty: Easy
Acceptance Rate: 74.31%
Paid Only: No
Tags: Array, Matrix

Problem Description

Given a `m x n` binary matrix `mat`, find the **0-indexed** position of the row that contains the **maximum** count of **ones,** and the number of ones in that row.

In case there are multiple rows that have the maximum count of ones, the row with the **smallest row number** should be selected.

Return _an array containing the index of the row, and the number of ones in it._

Example 1:

Input: mat = [[0,1],[1,0]] **Output:** [0,1] **Explanation:** Both rows have the same number of 1's. So we return the index of the smaller row, 0, and the maximum count of ones (1). So, the answer is [0,1].

Example 2:

Input: mat = [[0,0,0],[0,1,1]] **Output:** [1,2] **Explanation:** The row indexed 1 has the maximum count of ones (2). So we return its index, 1, and the count. So, the answer is [1,2].

Example 3:

Input: mat = [[0,0],[1,1],[0,0]] **Output:** [1,2] **Explanation:** The row indexed 1 has the maximum count of ones (2). So the answer is [1,2].

Constraints:

* `m == mat.length` * `n == mat[i].length` * `1 <= m, n <= 100` * `mat[i][j]` is either `0` or `1`.

Code Snippets

C++:

```
class Solution {
public:
vector<int> rowAndMaximumOnes(vector<vector<int>>& mat) {

}
};
```

Java:

```
class Solution {
public int[] rowAndMaximumOnes(int[][] mat) {

}
}
```

Python3:

```
class Solution:
def rowAndMaximumOnes(self, mat: List[List[int]]) -> List[int]:
```