Given a m \* n matrix mat of *ones* (representing soldiers) and *zeros* (representing civilians), return the indexes of the k weakest rows in the matrix ordered from the weakest to the strongest.

A row ***i*** is weaker than row ***j***, if the number of soldiers in row ***i*** is less than the number of soldiers in row ***j***, or they have the same number of soldiers but ***i*** is less than ***j***. Soldiers are **always** stand in the frontier of a row, that is, always *ones* may appear first and then *zeros*.

**Example 1:**

**Input:** mat =

[[1,1,0,0,0],

[1,1,1,1,0],

[1,0,0,0,0],

[1,1,0,0,0],

[1,1,1,1,1]],

k = 3

**Output:** [2,0,3]

**Explanation:**

The number of soldiers for each row is:

row 0 -> 2

row 1 -> 4

row 2 -> 1

row 3 -> 2

row 4 -> 5

Rows ordered from the weakest to the strongest are [2,0,3,1,4]

**Example 2:**

**Input:** mat =

[[1,0,0,0],

 [1,1,1,1],

 [1,0,0,0],

 [1,0,0,0]],

k = 2

**Output:** [0,2]

**Explanation:**

The number of soldiers for each row is:

row 0 -> 1

row 1 -> 4

row 2 -> 1

row 3 -> 1

Rows ordered from the weakest to the strongest are [0,2,3,1]

**Constraints:**

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