Maximum Weight Node

Given a maze with **N** cells. Each cell may have multiple entry points but not more than one exit (i.e entry/exit points are unidirectional doors like valves).
You are given an array **Edge[]** of **N** integers, where **Edge[i]** contains the cell index that can be reached from cell **i** in one step. **Edge[i]** is **-1** if the ith cell doesn't have an exit.
The task is to find the cell with **maximum weight** (The weight of a cell is the sum of cell indexes of all cells pointing to that cell). If there are multiple cells with the maximum weight return the cell with highest index.

Note: The cells are indexed with an integer value from 0 to N-1. If there is no cell pointing to the ith cell then the weight of the i'th cell is zero.

Example 1:

```
Input:
N = 4
Edge[] = {2, 0, -1, 2}
Output: 2
Explanation:
1 -> 0 -> 2 <- 3
weight of 0th cell = 1
weight of 1st cell = 0
(because there is no cell pointing to the 1st cell)
weight of 2nd cell = 0+3 = 3
weight of 3rd cell = 0
There is only one cell which has maximum weight
(i.e 2) So, cell 2 is the output.
```

Example 2:

```
Input:
```

```
N = 1
Edge[] = {-1}
Output: 0
Explanation:
weight of 0th cell is 0.
There is only one cell so
cell 0 has maximum weight.
```

Your task:
You dont need to read input or print anything. Your task is to complete the function **maxWeightCell**() which takes the integer N denoting the number of cells and the array Edge[] as input parameters and returns the cell which has maximum weight. If there are multiple answers then return the cell with highest index.

Expected Time Complexity: O(N)
Expected Auxiliary Space: O(N)

Constraints:
$1 \le N \le 10^5$
$-1 \le Edge[i] < N$
Edge[i] != i