Given an array of integers arr, you are initially positioned at the first index of the array.

In one step you can jump from index i to index:

* i + 1 where: i + 1 < arr.length.
* i - 1 where: i - 1 >= 0.
* j where: arr[i] == arr[j] and i != j.

Return *the minimum number of steps* to reach the **last index** of the array.

Notice that you can not jump outside of the array at any time.

**Example 1:**

**Input:** arr = [100,-23,-23,404,100,23,23,23,3,404]

**Output:** 3

**Explanation:** You need three jumps from index 0 --> 4 --> 3 --> 9. Note that index 9 is the last index of the array.

**Example 2:**

**Input:** arr = [7]

**Output:** 0

**Explanation:** Start index is the last index. You don't need to jump.

**Example 3:**

**Input:** arr = [7,6,9,6,9,6,9,7]

**Output:** 1

**Explanation:** You can jump directly from index 0 to index 7 which is last index of the array.

**Example 4:**

**Input:** arr = [6,1,9]

**Output:** 2

**Example 5:**

**Input:** arr = [11,22,7,7,7,7,7,7,7,22,13]

**Output:** 3

**Constraints:**

* 1 <= arr.length <= 5 \* 10^4
* -10^8 <= arr[i] <= 10^8