Warehouse Shipping Optimization

📜 Problem Description

The manager of an Amazon warehouse needs to ship **n** products from different locations. The location of the *i*-th product is represented by an array **locations[i]**. The manager is allowed to perform one operation at a time to ship the products.

Note: After shipping a product, it gets removed from the inventory, and the rest of the products which are currently not shipped come together keeping the order the same as before.

The goal is to find the **minimum number of operations** that the manager has to perform to ship all of the products.

⚙️ Operations

The manager can perform one of the following two operations:

1. If the inventory has two or more products, the manager can pick two products **x** and **y** from the inventory if they have different locations, i.e., *locations[x] != locations[y]*, and ship both of them.
2. If the inventory has one or more products, the manager can pick one product **x** from the inventory and ship it.

📝 Example

Given n=5 and locations = [1, 8, 6, 7, 7].

Operation	x	y	locations
1.	1	5	[8, 6, 7]
2.	1	3	[6]
3.	1	-	[]

The manager needs to perform **3 operations** to ship all of the products.

📋 Function Description

Complete the function *minOperation* in the editor.

Parameters:

int locations[n]: the location of each product.

Returns:

int: the minimum number of operations that the manager has to perform to ship all of the products.

🔒 Constraints

- 1 <= n <= 10^5
- 1 <= locations[i] <= 10^9