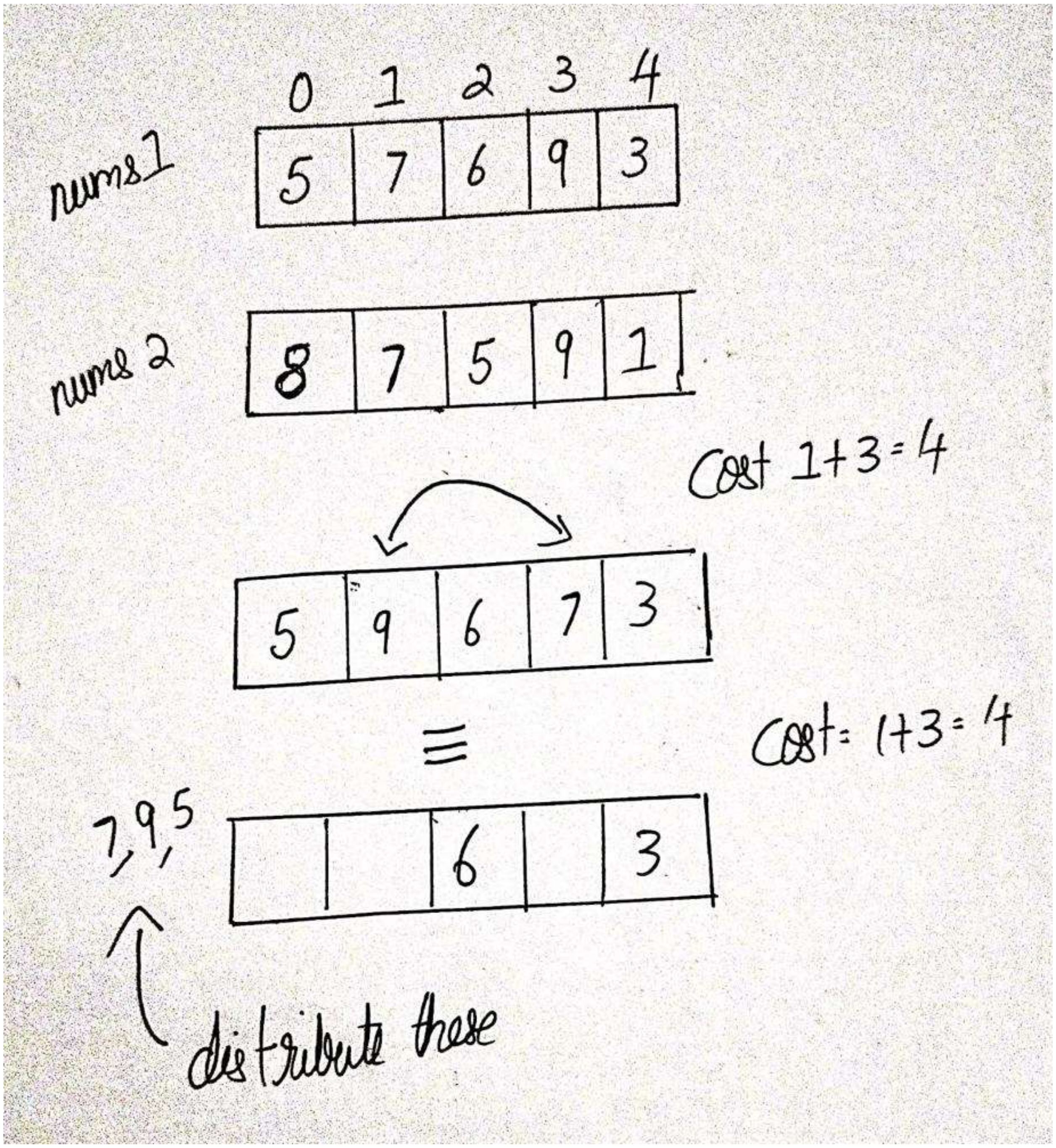
Intuition

1. The first position is like a resting place. The cost of moving an element from there is 0 as the index is 0.
2. Swapping 2 elements can be thought of as swapping both of the elements to position 0 and then swapping them back to desired position.



Approach

1. Lets call an index bad is nums1[index] = nums2[index]. Imagine this, you collect all "**bad indexes**" in the 0th index. The cost to move this index to the 0th index is the index of the element.
2. Now, you can use these collected indexes and distribute them to the desired positions. (note that we don't have to add any extra cost here, we can assume that once an element has come to position 0, it is free to move)
3. The only thing to care is the case where the maximum occuring element occurs more than half the values. In this case we need to collect more indexes even though they aren't "**bad**".
4. We want to collect indexes which contribute less cost, i.e which occur earlier (smallest index) as well as taking care that the value doesn't equal to the max occuring element (worsening our situation) and also ensuring that nums2[index] != max occurent element

Complexity

- Time complexity: O(n)O(n)$O(n)$
- Space complexity: O(n)O(n)$O(n)$