Two arrays are called *similar* if one can be obtained from another by swapping at most one pair of elements in one of the arrays.

Given two arrays `a` and `b`, check whether they are *similar*.

## Example

- For `a = [1, 2, 3]` and `b = [1, 2, 3]`, the output should be `areSimilar(a, b) = true`.

  The arrays are equal, no need to swap any elements.

- For `a = [1, 2, 3]` and `b = [2, 1, 3]`, the output should be `areSimilar(a, b) = true`.

  We can obtain `b` from `a` by swapping `2` and `1` in `b`.
- For `a = [1, 2, 2]` and `b = [2, 1, 1]`, the output should be `areSimilar(a, b) = false`.

  Any swap of any two elements either in `a` or in `b`won't make `a` and `b` equal.

## Input/Output

- **[execution time limit] 0.5 seconds (cpp)**

- **[input] array.integer a**

  Array of integers.

  *Guaranteed constraints:*
  `3 ≤ a.length ≤ 10^5`,
  `1 ≤ a[i] ≤ 1000`.

- **[input] array.integer b**

  Array of integers of the same length as `a`.
  *Guaranteed constraints:*
  `b.length = a.length`,
  `1 ≤ b[i] ≤ 1000`.

- **[output] boolean**

  - `true` if `a` and `b` are similar, `false` otherwise.