Task description Statistics Try Task Out

Programming language: Java ▾

An array A consisting of N integers is given. We are looking for pairs of elements of the array that are equal but that occupy different positions in the array. More formally, a pair of indices (P, Q) is called *identical* if 0 ≤ P < Q < N and A[P] = A[Q]. The goal is to calculate the number of identical pairs of indices.

For example, consider array A such that:

```
  A[0] = 3
  A[1] = 5
  A[2] = 6
  A[3] = 3
  A[4] = 3
  A[5] = 5
```

There are four pairs of identical indices: (0, 3), (0, 4), (1, 5) and (3, 4). Note that pairs (2, 2) and (5, 1) are not counted since their first indices are not smaller than their second.

Write a function:

```
function solution(A);
```

that, given an array A of N integers, returns the number of identical pairs of indices.

If the number of identical pairs of indices is greater than 1,000,000,000, the function should return 1,000,000,000.

For example, given:

```
  A[0] = 3
  A[1] = 5
  A[2] = 6
  A[3] = 3
  A[4] = 3
  A[5] = 5
```

the function should return 4, as explained above.

Assume that:

- N is an integer within the range [0..100,000];
- each element of array A is an integer within the range [−1,000,000,000..1,000,000,000].

Complexity:

- expected worst-case time complexity is O(N*log(N));
- expected worst-case space complexity is O(N) (not counting the storage required for input arguments).