**Counting Pairs**

QUESTION DESCRIPTION

Given an integer k and a list of integers, count the number of distinct valid pairs of integers (a, b) in the list for which a + k = b. Two pairs of integers (a, b) and (c, d) are considered distinct if at least one element of (a, b) does not also belong to (c, d).

**Example**

n = 4

numbers = [1, 1, 1, 2]

k = 1

This array has two different valid pairs: (1, 1) and (1, 2). For k = 1, there is only 1 valid pair which satisfies a + k = b: the pair (a, b) = (1, 2).

**Function Description**

Complete the function countPairs in the editor below.

countPairs has the following parameter(s):

int numbers[n]: array of integers

int k: target difference

**Returns**

int: number of valid (a, b) pairs in the numbers array that have a difference of k

**Constraints**

* 2 ≤ n ≤ 2 × 10
* 0 ≤ numbers[i] ≤ 10 , where 0 ≤ i < n
* 0 ≤ k ≤109

**Input Format for Custom Testing**

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer n, the size of the array numbers.

Each of the next n lines contains an integer numbers[i] where 0 ≤ i < n.

The next line contains an integer k.

**Sample Case 0**

**Sample Input 0**

STDIN Function

6 → numbers[] size n = 6

1 → numbers[] = [ 1, 1, 2, 2, 3, 3]

1

2

2

3

3

1 → k = 1

**Sample Output 0**

2

**Explanation 0**

There are 2 valid pairs in numbers = [1, 1, 2, 2, 3, 3] for k = 1, a + 1 = b:

1. (1, 2)

2. (2, 3)

**Sample Case 1**

**Sample Input 1**

STDIN Function

6 → numbers[] size n = 6

1 → numbers[] = [ 1, 2, 3, 4, 5, 6 ]

2

3

4

5

6

2 → k = 2

**Sample Output 1**

4

**Explanation 1**

There are four valid pairs in [1, 2, 3, 4, 5, 6] for k = 2, a + 2 = b.:

1. (1, 3)

2. (2, 4)

3. (3, 5)

4. (4, 6)

**Sample Case 2**

**Sample Input 2**

STDIN Function

6 → numbers[] size n = 6

1 → numbers[] = [ 1, 2, 5, 6, 9, 10]

2

5

6

9

10

2 → k = 2

**Sample Output 2**

0

**Explanation 2**

No valid (a, b) pair exists in [1, 2, 5, 6, 9, 10] for k = 2, a + 2 = b, so the function returns 0.

**Hint 1**

Can you fix the value of a and find if there is a corresponding b efficiently? Trying using Hash Tables.

**Hint 2**

Multiple occurrences of the same integer does not contribute to the answer.

**Solution**

Concepts covered: Hash Tables

**Optimal Solution:**

Create a hash table and store all the elements that occur in the array. Now, we know that b = a + k, hence if we fix the value of a, we can determine the value of b. So, iterate in the hash table and fix the value of a. Check the hash table to find if a + k exists. If it does, we have found a valid pair.

def countPairs(numbers, k):

contains = {j : True for j in numbers}

ans = 0

for j in contains:

if j + k in contains:

ans += 1

return ans

**Error Handling**: There could be cases where the candidate tries to calculate the number of distinct pairs according to indices instead of according to values.

**CANDIDATE ANSWER**

Language used: Java 8

class Result {

/\*

\* countPairs function.

\*

\* return an INTEGER.

\* @param list of integer numbers, k as int

\*

\*/

public static int countPairs(List<Integer> numbers, int k) {

// Logic

}