

Full Name:

Hideki Ikeda

Email:

hidekiai+hackerrank@gmail.com

https://www.linkedin.com/in/hidekiai/

Test Name:

**Mock Test** 

Taken On:

14 Sep 2023 00:39:44 IST

Time Taken:

23 min 25 sec/ 25 min

Linkedin: Invited by:

Ankush

Invited on:

14 Sep 2023 00:38:48 IST

Skills Score:

Tags Score:

Algorithms 50/75

Core CS 50/75

Medium 50/75

Search 50/75

problem-solving 50/75

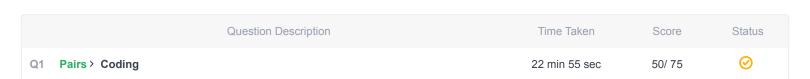
66.7%

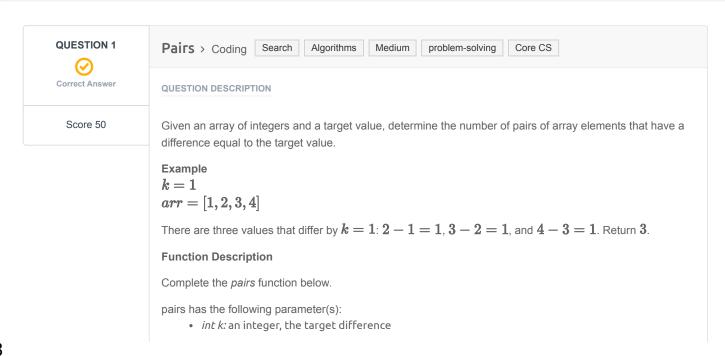
50/75

scored in **Mock Test** in 23 min 25 sec on 14 Sep 2023 00:39:44

### **Recruiter/Team Comments:**

No Comments.





• int arr[n]: an array of integers

#### Returns

int: the number of pairs that satisfy the criterion

#### **Input Format**

The first line contains two space-separated integers n and k, the size of arr and the target value. The second line contains n space-separated integers of the array arr.

#### **Constraints**

- $2 \le n \le 10^5$
- $0 < k < 10^9$
- $0 < arr[i] < 2^{31} 1$
- ullet each integer arr[i] will be unique

#### Sample Input

```
STDIN Function
-----
5 2 arr[] size n = 5, k =2
1 5 3 4 2 arr = [1, 5, 3, 4, 2]
```

## **Sample Output**

3

#### **Explanation**

There are 3 pairs of integers in the set with a difference of 2: [5,3], [4,2] and [3,1]. .

#### **CANDIDATE ANSWER**

# Language used: C++14

```
2 /*
   * Complete the 'pairs' function below.
4
   * The function is expected to return an INTEGER.
   * The function accepts following parameters:
   * 1. INTEGER k
8
   * 2. INTEGER ARRAY arr
   */
11 int pairs(int k, vector<int> arr) {
   // k=1
    // arr=[1,2,3,4]
14
   // Result=3: There are 3 values that differ by k=1:
   // 2-1=2, 3-2=1, and 4-3=1
    auto calc_difference = [&k] (int left, int right) -> bool {
     // cout << "k=" << k << ": " << left << "-" << right << "=" <<
     // abs(left-right) << " (" << (abs(left-right) == k) << ")" << endl;
     return abs(left - right) == k;
    };
    // NOTE: each integer arr[i] will be unique!
    // this means, ideally if the array was sorted, we may be able
    // to somehow determine the distance (k) and reduce the iteration
    // for optimization (later)
     // first attempt, just go from current and scan against all
```

```
// if we had 1,2,3,4 then we only need to evaluate 1 against 2,3,4
   // and 2 against 3,4 (because 1 has already been evaluated)
30 // and 3 against 4
    auto found count = 0;
   for (auto i = 0; i < arr.size(); ++i) {
     for (auto j = i + 1; j < arr.size(); ++j) {
        // because we can assume that each integers are unique, we can
        // probably just skip, but even if we did the diff, it'll be 0 so
       // it won't match (k > 0 \text{ if all integers are unique})
       if (calc_difference(arr[i], arr[j])) {
          ++found_count;
       }
     }
   }
   return found count;
43 }
44
```

TESTCASE	DIFFICULTY	TYPE	STATUS	SCORE	TIME TAKEN	MEMORY USED
Testcase 1	Easy	Hidden case	Success	5	0.0645 sec	8.98 KB
Testcase 2	Easy	Hidden case	Success	5	0.0597 sec	8.68 KB
Testcase 3	Easy	Hidden case	Success	5	0.0602 sec	8.75 KB
Testcase 4	Easy	Hidden case	Success	5	0.0568 sec	8.79 KB
Testcase 5	Easy	Hidden case	Success	5	0.0639 sec	8.97 KB
Testcase 6	Easy	Hidden case	Success	5	0.053 sec	9.05 KB
Testcase 7	Easy	Hidden case	Success	5	0.067 sec	9.11 KB
Testcase 8	Easy	Hidden case	Success	5	0.054 sec	9.07 KB
Testcase 9	Easy	Hidden case	Success	5	0.0459 sec	9.14 KB
Testcase 10	Easy	Hidden case	<b>⊘</b> Success	5	0.0894 sec	9.03 KB
Testcase 11	Easy	Hidden case	Terminated due to timeout	0	2.0022 sec	14.2 KB
Testcase 12	Easy	Hidden case	Terminated due to timeout	0	2.0138 sec	14.1 KB
Testcase 13	Easy	Hidden case	Terminated due to timeout	0	2.0023 sec	14.1 KB
Testcase 14	Easy	Hidden case	Terminated due to timeout	0	2.0028 sec	14.4 KB
Testcase 15	Easy	Hidden case	Terminated due to timeout	0	2.0027 sec	14.3 KB
Testcase 16	Easy	Sample case	<b>⊘</b> Success	0	0.0415 sec	8.99 KB
Testcase 17	Easy	Sample case	<b>⊘</b> Success	0	0.0312 sec	8.87 KB
Testcase 18	Easy	Sample case	Success	0	0.0706 sec	8.71 KB

No Comments