

Experiment 4 (Searching and Sorting)

Student Name: Bhanu Pundir

UID: 20BCS1439

Branch: BE CSE

Section/Group: 20BCS-WM_620B

Semester: 5th

Date of Performance: 24.08.2022

Subject Name: Competitive Coding

Subject Code: 20CSP_314

1. Aim/Overview of the practical:

HackerLand National Bank has a simple policy for warning clients about possible fraudulent account activity. If the amount spent by a client on a particular day is greater than or equal to $2 \times$ the client's **median** spending for a trailing number of days, they send the client a notification about potential fraud. The bank doesn't send the client any notifications until they have at least that trailing number of prior days' transaction data.

Given the number of trailing days d and a client's total daily expenditures for a period of n days, determine the number of times the client will receive a notification over all n days.

Example

expenditure = [10, 20, 30, 40, 50]

$d = 3$

On the first three days, they just collect spending data. At day 4, trailing expenditures are [10, 20, 30]. The median is 20 and the day's expenditure is 40. Because $40 \geq 2 \times 20$, there will be a notice. The next day, trailing expenditures are [20, 30, 40] and the expenditures are 50. This is less than 2×30 so no notice will be sent. Over the period, there was one notice sent.

Note: The median of a list of numbers can be found by first sorting the numbers ascending. If there is an odd number of values, the middle one is picked. If there is an even number of values, the median is then defined to be the average of the two middle values. ([Wikipedia](#))

Function Description

Complete the function activityNotifications in the editor below.

activityNotifications has the following parameter(s):

- int expenditure[n]: daily expenditures
- int d: the lookback days for median spending

- a. Fraudulent Activity Notifications
- b. Pairs

2. Task to be done/ Which logistics used:

- a. Fraudulent Activity Notifications



Returns

- int: the number of notices sent

Input Format

The first line contains two space-separated integers n and d , the number of days of transaction data, and the number of trailing days' data used to calculate median spending respectively.

The second line contains n space-separated non-negative integers where each integer i denotes $expenditure[i]$.

Constraints

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq d \leq n$
- $0 \leq expenditure[i] \leq 200$

b. Pairs

Given an array of integers and a target value, determine the number of pairs of array elements that have a difference equal to the target value.

Example

$k = 1$

$arr = [1, 2, 3, 4]$

There are three values that differ by $k = 1$: $2 - 1 = 1$, $3 - 2 = 1$, and $4 - 3 = 1$. Return 3.

Function Description

Complete the pairs function below.

pairs has the following parameter(s):

- int k : an integer, the target difference
- 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

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

3. Steps for Experiment/Practical/Code

Question 1: Fraudulent Activity Notifications

```
#include <bits/stdc++.h> using
namespace std; int
countArr[201];

float getMedian(int d){ int count
=0;
for(int i=0; i<201; i++){
    count+=countArr[i]; if(count*2>d){
        return(i/1.0);
    }
    else if(count *2==d){ for(int j=i+1;
        j<201; j++){ if(countArr[j]){ return
        (i+j)/(2.0);
        }
    }
}
return -1;
}

int activityNotifications(vector<int> expenditure, int d) { int notify = 0;
for(int i=0; i<expenditure.size(); i++){ if(i>=d){
    if(expenditure[i]>=2*getMedian(d)){ notify++;
    }
    countArr[expenditure[i-d]]--;
    }
    countArr[expenditure[i]]++;
}
return notify;
}

int main(){ int n, d; cin>>n>>d;
vector<int> v1(n); for(int i=0; i<n;
i++){ cin>>v1[i];
}
cout<<activityNotifications(v1, d); return 0;
}
```

Question 2: Pairs

```
#include <bits/stdc++.h> using
namespace std;
int pairs(vector < int > a,int k) { int ans = 0;
    map<int, int>    nums;    for(int
        i=0;i<a.size();i++) { nums[a[i]] = 1;
    }
    for(int i=0;i<a.size();i++) { if ( nums.find(a[i]-k) ==
nums.end() ) {
        } else { ans++;
        } }
    return ans; }

int main() { int res;

    int    _a_size,_k;    cin    >>    _a_size>>_k;    cin.ignore
(numeric_limits<streamsize>::max(), '\n'); vector<int> _a; int _a_item;
    for(int _a_i=0; _a_i<_a_size; _a_i++) { cin >> _a_item;
        _a.push_back(_a_item); }

    res = pairs(_a,_k); cout << res;
return 0;
}
```

5. Observations/Discussions/ Complexity Analysis:

- a. Fraudulent Activity Notifications
- b. Pairs.

6. Result/Output/Writing

Question 1: Fraudulent Activity Notifications

Congratulations!
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0	Input (stdin)	Download
1	9 5	
2	2 3 4 2 3 6 8 4 5	

Your Output (stdout)

1	2
---	---

Expected Output

1	2
---	---

Download

Question 2: Pairs

Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ Sample Test case 1

✓ Sample Test case 2

Input (stdin)

[Download](#)

```
1 5 2
2 1 5 3 4 2
```

Your Output (stdout)

```
1 3
```

Expected Output

[Download](#)

```
1 3
```