

Experiment 4

Student Name: Anjali Singh

Branch: CSE

Semester: 5th

Subject Name: CC Lab

UID: 20BCS9239

Section/Group:607/A

Date of Performance:28/08/22

Subject Code: 21CSP-314

- 1. Aim/Overview of the practical:** To implement the concept of searching and sorting.
- 2. Task to be done/ Which logistics used:**

In this practical we are going to understand various problems and find out a better approach to solve a particular problem related to searching and sorting on hackerrank.

a) Missing Numbers

Given two arrays of integers, find which elements in the second array are missing from the first array.

Example

arr =

[7,2,5,3,5,3] brr

=

[7,2,5,4,6,3,5,3]

The brr array is the original list. The numbers missing are [4,6].

Notes

If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number.

Return the missing numbers sorted ascending.

Only include a missing number once, even if it is missing multiple times.

The difference between the maximum and minimum numbers in the original list is less than or equal to 100.

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

3. Steps for experiment/practical/Code:

A) Missing Numbers

```
vector<int> missingNumbers(vector<int> arr, vector<int> brr)
{
    map<int, int> arr_1;
    map<int, int> brr_1;
    for (int i = 0; i < arr.size(); i++)
    {
        arr_1[arr[i]]++;
    }
    for (int i = 0; i < brr.size(); i++)
    {
        brr_1[brr[i]]++;
    }
    vector<int> ans;
    map<int, int>::iterator j = arr_1.begin();
    for (auto i : brr_1)
    {
        if (i.first == j->first)
        {
            if (j != arr_1.end())
            {
                if (i.second == j->second)
                {
                    j++;
                }
                else
                {
                    ans.push_back(i.first);
                    j++;
                }
            }
        }
        else
        {
            ans.push_back(i.first);
        }
    }
    return ans;
}
```

```
}  
}
```

B) Pairs.

```
int pairs(int k, vector<int> arr)  
{  
    unordered_map<int, int> m;  
    int count = 0;  
    for (int i = 0; i < arr.size(); i++)  
    {  
        m[arr[i]]++;  
    }  
    for (int i = 0; i < arr.size(); i++)  
    {  
        if (m[arr[i] + k])  
        {  
            count++;  
        }  
    }  
    return count;  
}
```

5. Observations/Discussions/ Complexity Analysis:

- a). In missingNumbers function we are comparing two vectors arr and brr and trying to find missing values that are present in brr but not in arr. Also we are checking to ensure that If a number occurs multiple times in the lists, you must ensure that the frequency of that number in both lists is the same. If that is not the case, then it is also a missing number. We are applying sorting technique here.
- b) In pairs we have 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. We are using

unordered map in this question. And in the end we are returning the number of pairs that satisfy the criterion.

6. Result/Output/Writing Summary:

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**

Input (stdin)		Download
1	10	
2	203 204 205 206 207 208 203 204 205 206	
3	13	
4	203 204 204 205 206 207 205 208 203 206 205 206 204	

Your Output (stdout)	
1	204 205 206

Expected Output		Download
1	204 205 206	

Congratulations

You solved this challenge.
Would you like to challenge your friends?

[Next Challenge](#)

Earn a certificate in Problem Solving

Kudos on your progress! Take the HackerRank Skills Certification test and enrich your profile

[Get Certified](#)

- Test case 0
- Test case 1
- Test case 2
- Test case 3
- Test case 4
- Test case 5
- Test case 6

Compiler Message

Success

Input (stdin)

```

1 226 687
2 967551 42711 652888 556131 432461 689084 878716 707516 462627
719131 921983 626364 4162 381455 628368 434733 845482 789342
129922 384203 516975 872544 958157 257969 383516 972027 753530
579132 732314 692990 938898 673695 304274 911598 386363 643777
897942 705604 307205 691832 525153 13560 131633 967903 704917
719007 275998 823133 381356 694812 130946 14247 881464 212321
535615 388113 263786 993408 303132 347246 957012 356253 80278
682192 79591 22168 399471 130259 302213 146877 143512 464001

```

[Download](#)

HackerRank

PREPARE CERTIFY COMPETE

Search

hardhatDay

All Submissions

Submissions

Sort by Date Sort by Challenge

Problem	Language	Time	Result	Score
Pairs	C++14	2 minutes ago	Accepted	50.0
Balanced Brackets	C++20	about 3 hours ago	Accepted	25.0
Game of Two Stacks	C++20	about 3 hours ago	Accepted	30.0
Game of Two Stacks	C++20	7 days ago	Accepted	30.0
Down to Zero II	C++20	7 days ago	Abort Called	28.8
Down to Zero II	C++20	7 days ago	Abort Called	28.8
Down to Zero II	C++20	7 days ago	Abort Called	28.8
Balanced Brackets	C++20	8 days ago	Accepted	25.0
Game of Two Stacks	C++20	8 days ago	Accepted	30.0
Equal Stacks	C++20	8 days ago	Accepted	25.0

1 2 3 4 5 6 7 8 9 10

Learning outcomes (What I have learnt):

1. I have learnt how to use different functions and library of c++.

2. I have learnt how to deal with real time problems.
3. Both questions help me to build different logic and concept.
4. Learnt how to implement searching and sorting and do various types of functions with it.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			