

## Experiment Title 4 (Searching and Sorting)

### 1. Aim/Overview of the practical: *Fraudulent Activity Notifications*

### 2. Task to be done:

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

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

$d = 3$

### 3. Code:

```
#include <iostream>
#include <vector>
#include <map>
#include <set>
#include <algorithm>

using namespace std;

#define MAXE 210

int A[200010];
int F[MAXE];

int median2(int D) {
    int p = 0;
    for (int i = 0; i < MAXE; i++) {
        p += F[i];
```

```
if (p * 2 > D) {  
    return 2 * i;  
} else if (p * 2 == D) {  
    for (int j = i + 1; ; j++) {  
        if (F[j]) {  
            return i + j;  
        }  
    }  
}  
return -1;  
}
```

```
int main() {  
    int N, D;  
    //(Jitesh Kumar 20BCS2334)  
    cin >> N >> D;  
    for (int i = 0; i < N; i++) {  
        cin >> A[i];  
    }  
    int result = 0;  
    for (int i = 0; i < N; i++) {  
        if (i >= D) {  
            if (A[i] >= median2(D)) {  
                ++result;  
            }  
            F[A[i] - D]--;  
        }  
        F[A[i]]++;  
    }  
    cout << result << endl;
```

```
return 0;  
}
```

## 4. Output:

### 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	9 5
2	2 3 4 2 3 6 8 4 5

Your Output (stdout)

1	2
---	---

Expected Output

[Download](#)

1	2
---	---

## 4.2 Aim/Overview of the practical: Quicksort 1 - Partition

### 1. Task to be done:

Choose some pivot element,  $p$ , and partition your unsorted array,  $arr$ , into three smaller arrays:  $left$ ,  $right$ , and  $equal$ , where each element in  $left < p$ , each element in  $right > p$ , and each element in  $equal = p$ .

#### Example

$arr = [5, 7, 4, 3, 8]$

In this challenge, the pivot will always be at  $arr[0]$ , so the pivot is 5.

$arr$  is divided into  $left = \{4, 3\}$ ,  $equal = \{5\}$ , and  $right = \{7, 8\}$ .

Putting them all together, you get  $\{4, 3, 5, 7, 8\}$ . There is a flexible checker that allows the elements of  $left$  and  $right$  to be in any order. For example,  $\{3, 4, 5, 8, 7\}$  is valid as well.

Given  $arr$  and  $p = arr[0]$ , partition  $arr$  into  $left$ ,  $right$ , and  $equal$  using the Divide instructions above. Return a 1-dimensional array containing each element in  $left$  first, followed by each element in  $equal$ , followed by each element in  $right$ .

### 2. Code:

```
#include <map>
#include <set>
#include <list>
#include <cmath>
#include <ctime>
#include <deque>
#include <queue>
#include <stack>
#include <bitset>
#include <cstdio>
#include <vector>
#include <cstdlib>
#include <numeric>
#include <sstream>
#include <iostream>
#include <algorithm>
using namespace std;
```

```
void swap(vector<int> &arr, int i, int j){
    if(i==j) return;
    int tmp = arr[i];
    arr[i] = arr[j];
```

```
arr[j] = tmp;
}

/* Head ends here */
void partition(vector<int> ar) {
    int boundary = 0, i = 1, number_of_ele = ar.size(), tmp = 0;
    while(i < number_of_ele) {
        if(ar[i] < ar[0]) {
            tmp = ar[i];
            for(int j = i; j > boundary + 1; j--) {
                ar[j] = ar[j - 1];
            }
            boundary++;
            ar[boundary] = tmp;
        }
        i++;
    }
    int pivot = ar[0];
    for(i = 0; i < boundary; i++)
        ar[i] = ar[i + 1];
    ar[boundary] = pivot;
    for(i = 0; i < number_of_ele; i++) {
        if(i < number_of_ele - 1) cout << ar[i] << " ";
        else cout << ar[i] << endl;
    }
}

/* Tail starts here */
int main() {
    //(Jitesh Kumar 20BCS2334)
    vector<int> _ar;
    int _ar_size;
    cin >> _ar_size;
    for(int _ar_i = 0; _ar_i < _ar_size; _ar_i++) {
        int _ar_tmp;
        cin >> _ar_tmp;
        _ar.push_back(_ar_tmp);
    }

    partition(_ar);

    return 0;
}
```

### 3. Output:

#### 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

Compiler Message

Success

Input (stdin)

[Download](#)

1	5
2	4 5 3 7 2

Your Output (stdout)

1	3 2 4 5 7
---	-----------

Expected Output

[Download](#)

1	3 2 4 5 7
---	-----------