

Experiment No. 2

Title: Implementation of problem on competitive programming platform and optimizing it using array and hash table data structures



Batch:B1 Roll No:16010420133 Experiment No.:2

Aim: To implement a problem on hacker earth and optimize it's solution using array and hash table data structures

Resources needed: Web Browser to access Hackerearth platform

Theory:

Competitive Programming involves solving coding problems using data structures and algorithms. Competitive programming helps to improve logical and analytical skills. There are various platforms available for Competitive Programming such as Hackerearth, Codechef, Codeforces etc. On Competitive Programming platform, one needs to write solution under various restrictions such as memory limits, execution time, constraints on input size and so on. The Competitive Programming Platform then evaluates the solution against pre-defined test cases for a problem statement.

Hackerearth is a competitive programming platform which hosts programming challenges and coding competitons. On Hackerearth platform, the problem statement is defined in terms of real world scenario followed by input format, output format, constraints, sample input, sample output, time limit in seconds and memory limit. Hackerearth supports several programming languages such as C, C++, Java, Python, JavaScript and so on. Any of these supported programming languages can be selected for the implementation of the solution. There is a code editor provided where the code can be written and compiled. When the solution is submitted on the hackerearth platform, it is tested against pre-defined test cases for the problem statement. And it displays the list of all test cases and the status of solution against each test case whether solution has passed that test case or not.

Solving problem on Competitive Programming Platform like hackerearth helps to:

- 1. Understand problem in terms of real-world scenario
- 2. Interpret input, output and processing information and constraints from the given real world scenario problem
- 3. Develop logical and analytical ability for optimization of solution
- 4. Understand and perform test-case based evaluation of solution

Activity:

Consider the following problem statement. Solve it using array and hash table data structures on hackerearth platform

There are N numbers A1,A2,A3,...,AN, and you are given Q queries. In each query, you are given two integers L and R.

You are required to print the sum of all the numbers whose frequency of occurrence is between L and R (including L and R). Print a single integer for each query in a new line.

Input format

- The first line contains N denoting the size of the array.
- The second line contains N integers denoting the elements of the array.
- The third line contains Q denoting the number of queries. Next Q lines contain L and R

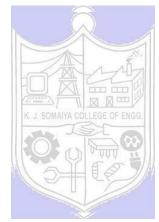
Output format

For each query, print the sum of all frequency of occurrence is between



$$\begin{split} &1 \leq N \leq 10^6 \\ &1 \leq Ai \leq 10^6 \ 1 \leq Q \leq 10^6 \\ &1 \leq l \leq r \leq N \end{split}$$





elements of the array whose L and R (inclusive) in a new line.

Sample Output

37

17

9

0

Program:

```
# Write your code
here N = int(input())
xs = input()
xs = [int(x) for x in]
xs.split()] Q = int(input()) qs
= [] for i in range(Q):
input()
           [l, r] = q.split()
                                                    int(r)))
    qs.append((int(1),
# number:frequency
(dictionary) nf = {} for x in
xs:
    f = nf.get(x, 0)
                         nf[x]
= f + 1
# frequency:sum of numbers of
same frequency dictionary fn
                                                     = {}
for k,v in nf.items():
                fn[v] = ns + k*v
fn.get(v, 0)
# list with its index as the frequency and the value as the sum of numbers of
this frequency fs = [] for i in range(max(fn.keys()) + 1):
    fs.append(fn.get(i, 0))
# solution for
(1, r) in qs:
    print(sum(fs[l:r+1]))
```

Output:

KJSCE/IT/SYBTECH/SEM IV/CPL/2021-22



Outcomes:	Understand	the	fundamental	concepts	for	managing	the	data	using	different	data
structures such as lists, queues, trees etc											

Conclusion: (Conclusion to be based on the objectives and outcomes achieved) Successfully solved the given problem statement and understood the concepts related to data structures

References:

- 1. https://www.hackerearth.com/practice/data-structures/arrays/1-d/practice-problems/algorithm/sum-as-per-frequency-88b00c1f/
- 2. T.H. Coreman ,C.E. Leiserson,R.L. Rivest, and C. Stein, "Introduction to algorithms", 3rd Edition 2009, Prentice Hall India Publication
- 3. Antti Laaksonen, "Guide to Competitive Programming", Springer, 2018
- 4. Gayle Laakmann McDowell," Cracking the Coding Interview", CareerCup LLC, 2015
- 5. Steven S. Skiena Miguel A. Revilla,"Programming challenges, The Programming Contest Training Manual", Springer, 2006
- 6. Antti Laaksonen, "Competitive Programmer's Handbook", Hand book, 2018
- 7. Steven Halim and Felix Halim, "Competitive Programming 3: The Lower Bounds of Programming Contests", Handbook for ACM ICPC