

Problem Statement:

Problem

There are N numbers A_1, A_2, \dots, A_n , 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 elements of the array whose frequency of occurrence is between l and r (inclusive) in a new line.

Constraints

$$1 \leq N \leq 1000000$$

$$1 \leq A_i \leq 1000000$$

$$1 \leq Q \leq 1000000$$

$$1 \leq l \leq r \leq N$$

Sample Input	Sample Output
8 4 4 6 5 3 3 3 9 4 1 4 2 7 3 7 5 6	37 17 9 0

Time Limit: 1

Memory Limit: 256

Explanation

In the first query we need to output the sum of all the numbers whose frequency of occurrence is between 1 and 4 (inclusive). Here all the given numbers have their frequency between 1 and 4. 3 occurs 3 times, 4 occurs 2 times, 5, 6, 9 occurs 1 time. So the total sum would be $3+3+3+4+4+5+6+9 = 37$.

Working:

25 / 01 / 2024

I/O

- $N \rightarrow$ size of array
- N integers in single line, space-separated [array elements]
- $Q \rightarrow$ no of queries
- l and r , $\frac{l \leftrightarrow r}{\text{space}}$ } Q lines

O/P

- Sum of all elements with frequency $\in [l, r]$ } Q lines

Time limit exceeded!

Approach:

$N, A, Q \rightarrow$ I/O ✓ # $O(3)$

frequency-map = Counter(A)

→ from collections lib

```
query-result = []
for i in range(Q):
    l, r = input() # O(1)
    sum = 0
    for element, freq in frequency_map.items():
        if element >= l & element <= r:
            sum += element * freq # O(1)
    query_result.append(sum) # O(1)

for i in query_result:
    print(i) # O(Q)
```

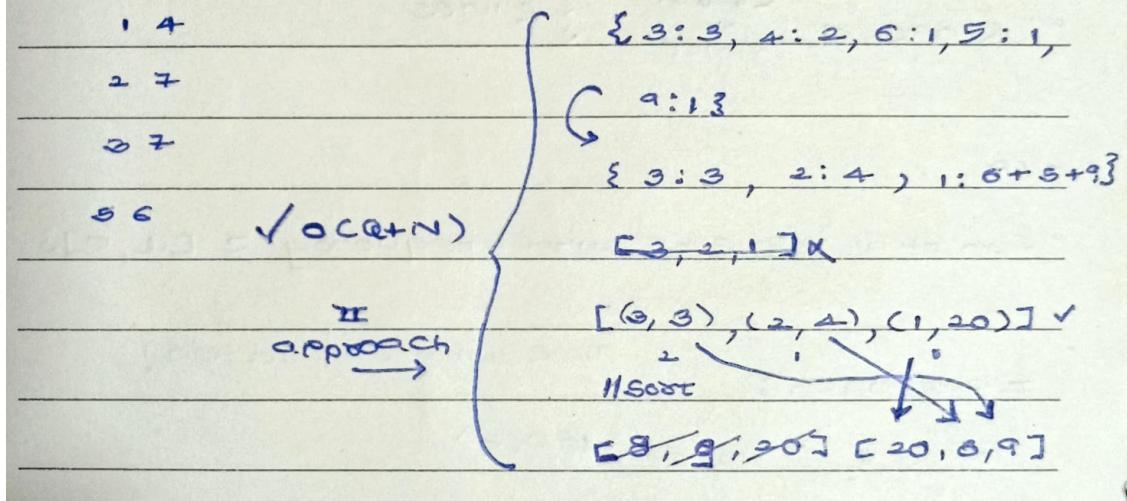
25 / 01 / 2024

Final time complexity

$$= O(3) + O(N) + O(1) + O(Q \times (4 + O(N))) + O(Q)$$

$$= O(N + 5Q + QN + 4)$$

$$= O(N + Q + QN) \quad !$$



Code:

```
from collections import Counter
from math import prod
N = int(input())
A = list(map(int, input().split()))
Q = int(input())
frequency_map = Counter(A)
k = {}
result = []
for element, frequency in frequency_map.items():
    if frequency in k:
        k[frequency] += element
    else:
        k[frequency] = element
```

```

freq_sorted = sorted(list(k.items()))
element_sum = list(map(prod, freq_sorted))
for i in range(Q):
    l, r = list(map(int, input().split()))
    result.append(sum(element_sum[l-1 : r]))
for i in result:
    print(i)

```

```

PS C:\Users\chand\Downloads\IV SEM\CPL>
_per_frequency.py"
8
4 4 6 5 3 3 3 9
4
1 4
2 7
3 7
5 6
37
17
9
0

```

Result:

RESULT: Accepted

[Refer judge environment](#)

Score	Time (sec)	Memory (KiB)	Language
30	9.97929	127852	Python 3.8

Input	Result	Time (sec)	Memory (KiB)	Score	Your output	Correct output	Diff
Input #1	Accepted	0.026521	3264	10			
Input #2	Accepted	1.31856	9120	10			
Input #3	Accepted	0.792302	127852	10			
Input #4	Accepted	0.212565	9192	10			
Input #5	Accepted	0.243312	11988	10			
Input #6	Accepted	2.338327	90564	10			
Input #7	Accepted	0.527638	62244	10			
Input #8	Accepted	0.84142	17168	10			
Input #9	Accepted	1.914166	110728	10			