

Name:

Roll Number:

ESO207: Data Structures and Algorithms

Programming Assignment 2

Due Date: 19th February, 2021

Total Number of Pages: 5

Total Points 100

Note :

- All questions have to be answered through a contest in Hackerrank. The contest has 4 challenges, each corresponding to a part. You have to submit your code through the contest. Following is the link to the contest: <https://www.hackerrank.com/eso207a-pa2-2020-21-ii>
- Your codes will be checked for possible plagiarism of any sorts. If we find such cases, then we will possibly award an F grade.
- Allowed Languages for challenge code submission : C, C++
- Allowed libraries : `stdio.h` for C and `iostream` for C++
- Use the same hackerrank username for programming assignment 2 as you have used for programming assignment 1.
- You will also need to upload all your program files (C/C++) on moodle.

Name:

Rollno:

Question 1. (20 points) **Another minimum sum**

Description:

Given an array A consisting of N elements, find the smallest sum of contiguous elements that is strictly greater than K .

Input:

First line will contain a single number N , denoting the number of elements in the array A .

Second line will contain the N elements of the array.

Third line will contain the integer K .

Output:

Output a single integer S , the minimum sum of contiguous elements that is strictly greater than K . If no sum exists that is strictly greater K , then output -1 .

Constraints:

$$1 \leq N \leq 10^5$$

$$-10^6 \leq A[i] \leq 10^6$$

$$1 \leq k \leq 10^6$$

Example:

Sample Input:

```
5
1 -5 3 -7 8
3
```

Sample Output:

```
4
```

Explanation:

The contiguous sequence 3 -7 8 has the smallest sum strictly greater 3 i.e. 4.

Note: the sum of the sequence -5 3 -7 is -9 but its not strictly greater 3

Name:

Rollno:

Question 2. (30 points) **Finding products**

Description:

Given two sorted (in non decreasing order) arrays of positive integers A and B having sizes M and N respectively, and an integer K , find the K th smallest product $A[i] \cdot B[j]$, where $0 \leq i \leq M - 1$ and $0 \leq j \leq N - 1$.

Input:

First line contains two integers, M and N , denoting the sizes of the two arrays A and B respectively.

Next line contains M positive integers in non decreasing order, the elements of A

Next line contains N positive integers in non decreasing order, the elements of B

Last line contains the integer K

Output:

Output a single integer P that is the K th smallest product $A[i] \cdot B[j]$

Constraints:

$1 \leq N, M \leq 100000$

$1 \leq A[i], B[i] \leq 10000$

$1 \leq k \leq N * M$

Example:

Sample Input:

```
3 4
1 3 4
2 4 6 8
5
```

Sample Output:

```
8
```

Explanation:

The products $A[i] \cdot B[j]$ after sorting in non-decreasing order are as follows

```
2 4 6 6 8 8 12 16 18 24 24 32
```

The 5th smallest element in the above list is 8.

Name:

Rollno:

Question 3. (30 points) **Greater to the left**

Description:

Given an array A of N elements, Find for each element $A[i]$, how many elements $A[j]$ are greater than or equal to $A[i]$ such that $j < i$

Input:

First line contains an integer N , the number of elements in the array A .

Second line contains N elements, the contents of the array A .

Output:

Print N elements, where the i th element represents the number of elements $A[j]$ greater than or equal to $A[i]$

Constraints:

$1 \leq N \leq 500000$

$1 \leq A[i] \leq 100000$

Example:

Sample Input:

4
5 4 2 5

Sample Output:

0 1 2 1

Explanation:

There are 0 elements ≥ 5 to its left

There is 1 element ≥ 4 to its left i.e 5

There are 2 elements ≥ 2 to its left i.e. 5 and 4

There is 1 element ≥ 5 to its left i.e. 5

Name:

Rollno:

Question 4. (20 points) **Sonic and coins**

Description:

Sonic is busy collecting coins. He is traveling through a binary search tree having N nodes, where every node in the tree has one coin. When he travels from the node p to q he will collect all the coins in the path from p to q (including the coins in the nodes p and q).

Given K queries, where each query will have two numbers, p and q . Print one number, denoting the total number of coins Sonic will collect when traveling from the node p to q .

Input:

First line will contain N and K , denoting the number of nodes in the tree and number of queries respectively.

Next line will contain N integers, denoting the pre-order traversal of the nodes of the tree (all N numbers will be unique).

Next K lines will denote two integers p, q each, denoting the start and end node of the path that sonic will travel.

Output:

Print K numbers, where each number represents the number of coins Sonic will collect.

Constraints:

$$1 \leq N \leq 100000$$

$$1 \leq K \leq 100000$$

Example:

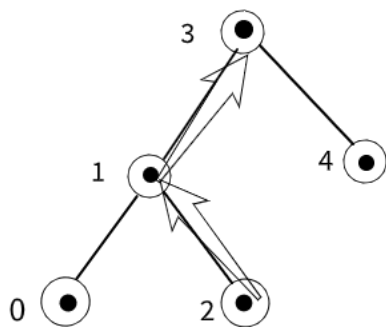
Sample Input:

```
5 2
3 1 0 2 4
2 3
0 4
```

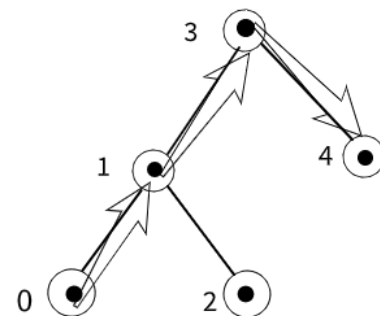
Sample Output:

```
3
4
```

Explanation:



(a) Sonic's first run, collects 3 coins.



(b) Sonic's second run, collects 4 coins.