Gmail YT Home CF LC Student Services TopicsIP Webmail CC C2Ladder MonthlyBills IITP

26m left

ALL

1

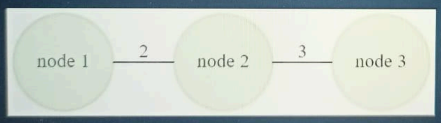
2

1. Bitwise Tree

Given a tree of *tree_nodes* vertices numbered from 1 to *tree_nodes*, each edge is associated with some weight, given by array *tree_weight*.
There are *q* queries in the form of a 2D array, *queries*, where each element consists of two integers (say *u* and *v*). For each query, compute the sums of the bitwise XOR, the bitwise AND, and the bitwise OR of the weights of the edges on the path from *u* to *v*.
Return the answers to each query as an array of integers.

Example
tree_nodes = 3
tree_from = [1, 3]
tree_to = [2, 2]
tree_weight = [2, 3]
q = 2
queries = [[1, 3], [2, 3]]

Edges of the tree in the form (u, v, w) are [[1, 2, 2], [3, 2, 3]].



- For *queries[0]*, *u = 1*, *v = 3*, XOR of weights = 2 ^ 3 = 1, AND of weights = 2 & 3 = 2, OR of weights = 2 | 3 = 3. Hence the answer to this query is 1 + 2 + 3 = 6.
- For *queries[1]*, *u = 2*, *v = 3*, XOR of weights = 3, AND of weights = 3, OR of weights = 3. Hence the answer to this query is 3 + 3 + 3 = 9.

Language C++20

Test Results

Compiled

Use print or lo cases are u cases.

Test case

Test case 1

Test case 2

Type here to search

Gmail YT Home CF LC Student Services TopicsIP Webmail CC C2Ladder MonthlyBills IITP

26m left

- For *queries[0]*, *u = 1*, *v = 3*, XOR of weights = 2 ^ 3 = 1, AND of weights = 2 & 3 = 2, OR of weights = 2 | 3 = 3. Hence the answer to this query is 1 + 2 + 3 = 6.
- For *queries[1]*, *u = 2*, *v = 3*, XOR of weights = 3, AND of weights = 3, OR of weights = 3. Hence the answer to this query is 3 + 3 + 3 = 9.

Return the array [6, 9].

Function Description
Complete the function *getPathXORAND* in the editor below.

getPathXORAND has the following parameters:
int tree_nodes: the number of nodes in the tree
int tree_from[tree_edges]: each *tree_from[i]* is one end of edge *i*
int tree_to[tree_edges]: each *tree_to[i]* is one end of edge *i*
int tree_weight[tree_nodes]: the weights of the edges
int queries[q][2]: a 2D array of queries

Returns
int[q]: the answers to the queries

Constraints

- $1 \le tree\_nodes \le 10^5$
- $1 \le q \le 10^5$
- $1 \le queries[i][0], queries[i][1], tree\_from[i], tree\_to[i] \le n$
- $1 \le tree\_weight[i] < 2^{20}$
- It is guaranteed that lengths of arrays *tree_from*, *tree_to*, *tree_weight* equal *tree_nodes* - 1.

▸ Input Format For Custom Testing

▾ Sample Case 0

Language C++20

Test Results

Compiled

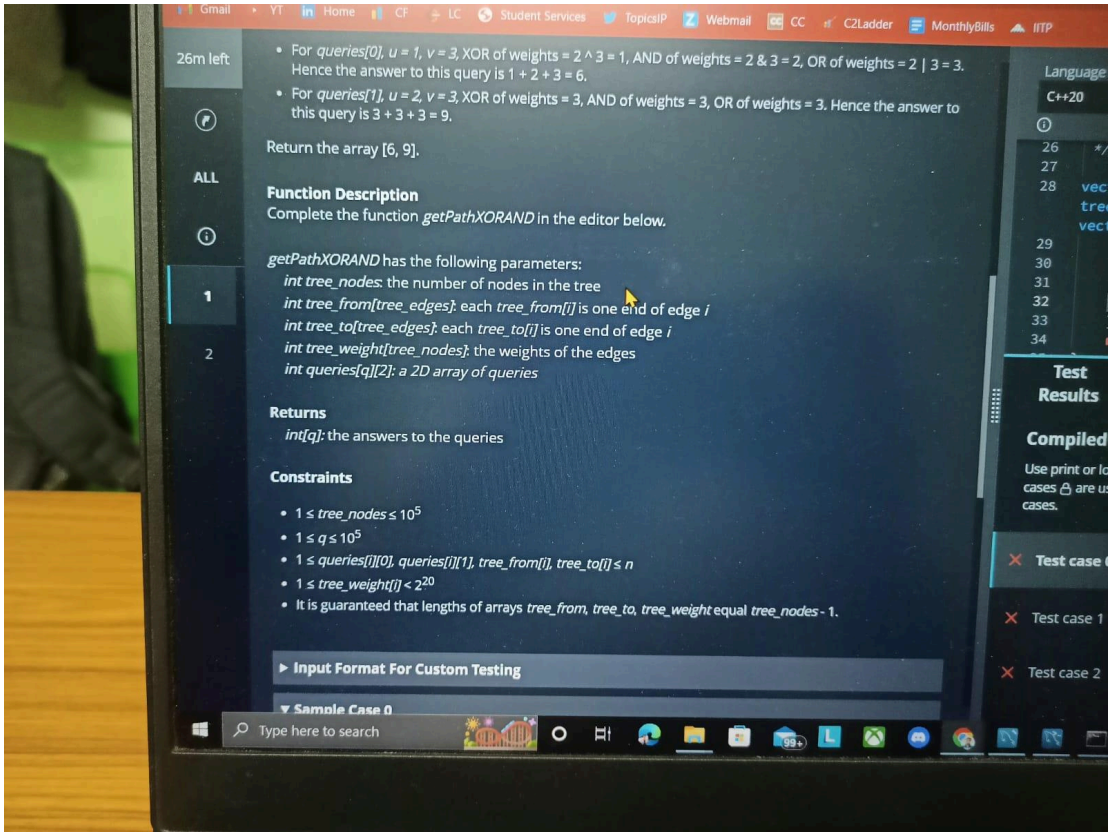
Use print or lo cases are u cases.

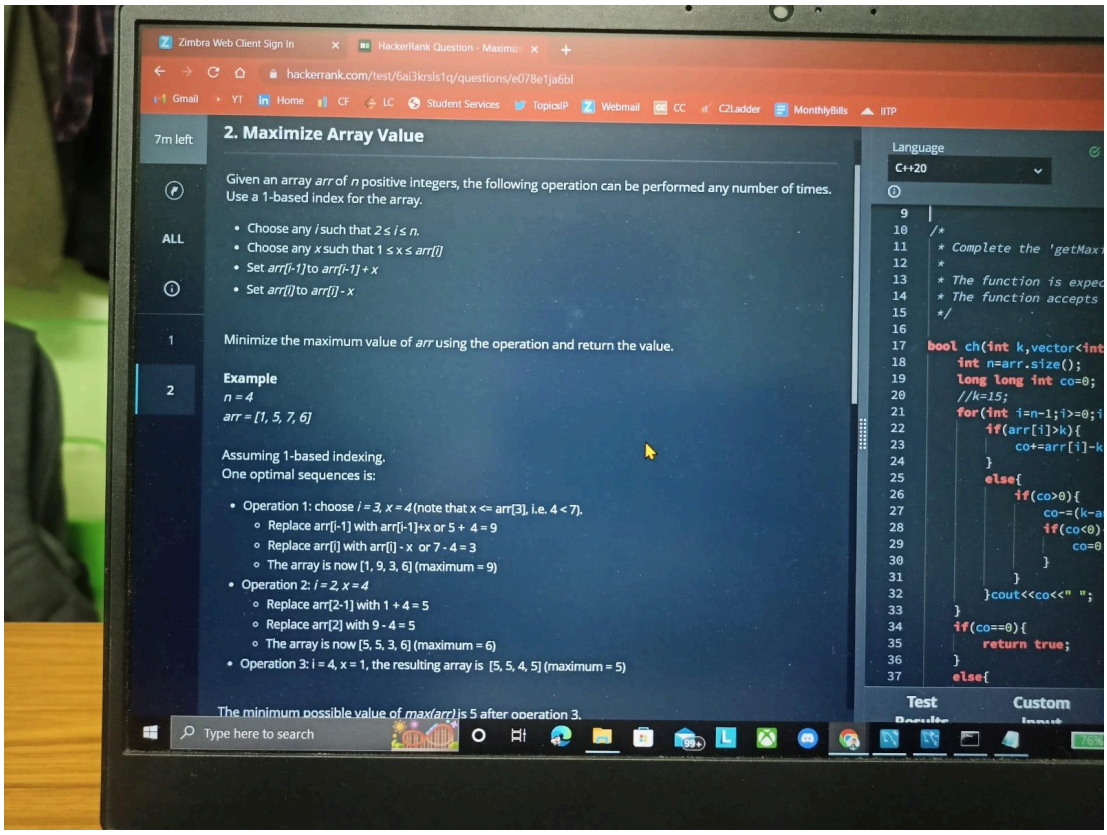
Test case

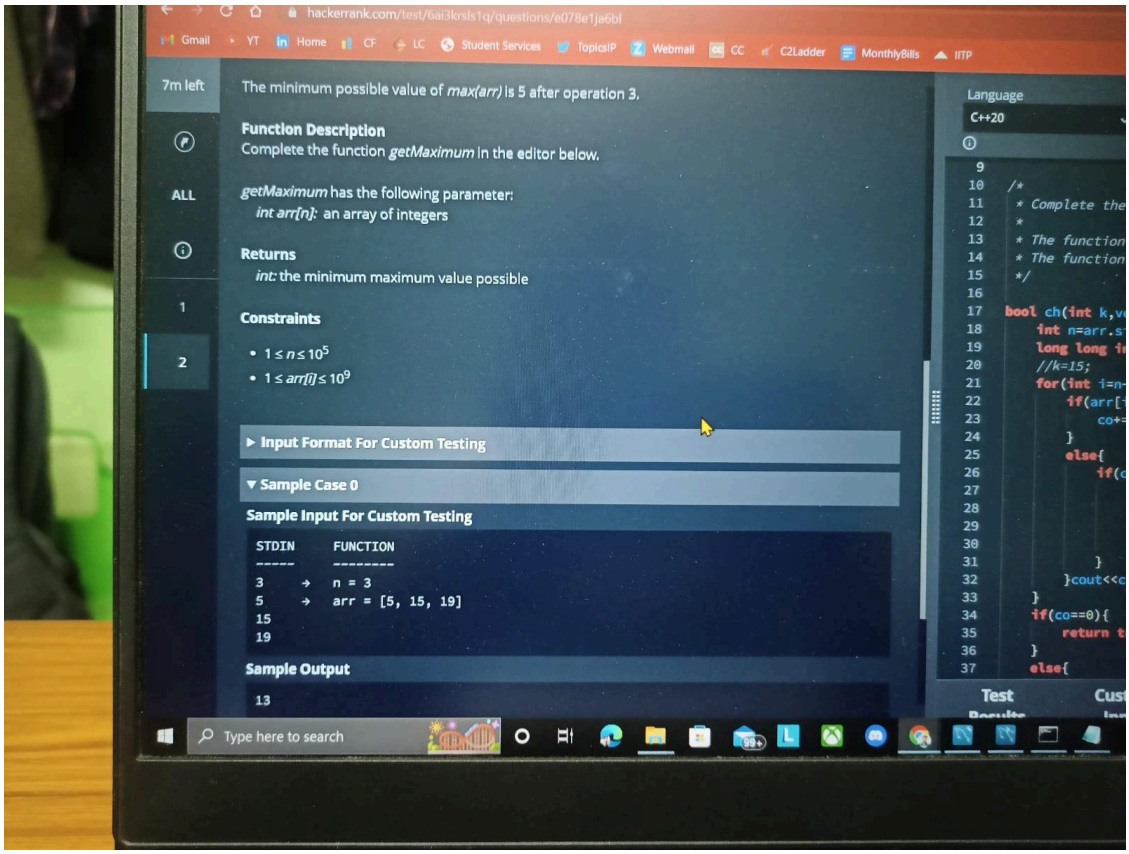
Test case 1

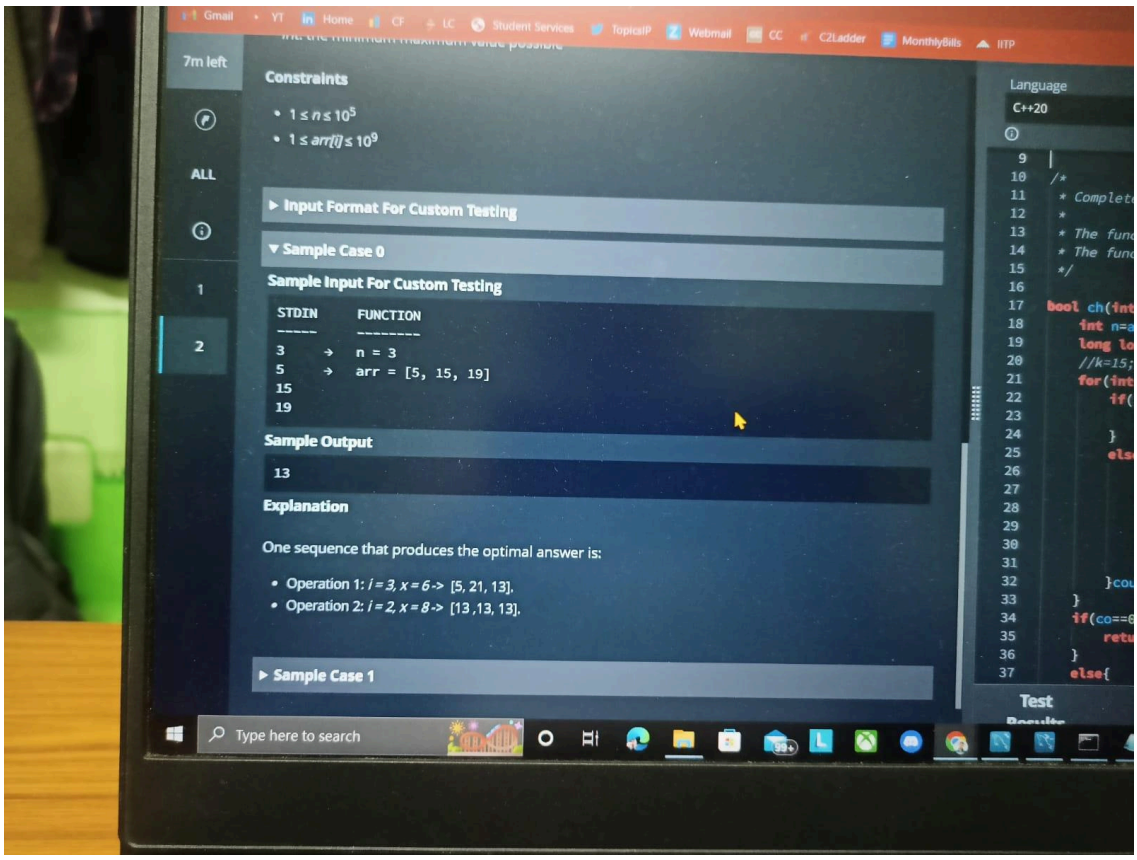
Test case 2

Type here to search


26m left
ALL
1
2
For queries[0], u = 1, v = 3, XOR of weights = 2 ^ 3 = 1, AND of weights = 2 & 3 = 2, OR of weights = 2 | 3 = 3. Hence the answer to this query is 1 + 2 + 3 = 6.
For queries[1], u = 2, v = 3, XOR of weights = 3, AND of weights = 3, OR of weights = 3. Hence the answer to this query is 3 + 3 + 3 = 9.
Return the array [6, 9].
Function Description
Complete the function getPathXORAND in the editor below.
getPathXORAND has the following parameters:
int tree_nodes: the number of nodes in the tree
int tree_from[tree_edges]: each tree_from[i] is one end of edge i
int tree_to[tree_edges]: each tree_to[i] is one end of edge i
int tree_weight[tree_nodes]: the weights of the edges
int queries[q][2]: a 2D array of queries
Returns
int[q]: the answers to the queries
Constraints
1 ≤ tree_nodes ≤ 10^5
1 ≤ q ≤ 10^5
1 ≤ queries[i][0], queries[i][1], tree_from[i], tree_to[i] ≤ n
1 ≤ tree_weight[i] < 2^20
It is guaranteed that lengths of arrays tree_from, tree_to, tree_weight equal tree_nodes - 1.
Input Format For Custom Testing
Sample Case 0
Language
C++20
Test Results
Compiled
Test case
Test case 1
Test case 2
Type here to search


Zimbra Web Client Sign In
HackerRank Question - Maximiz
hackerrank.com/test/6ai3krsls1q/questions/e078e1ja6bl
7m left
ALL
1
2
2. Maximize Array Value
Given an array arr of n positive integers, the following operation can be performed any number of times.
Use a 1-based index for the array.
Choose any i such that 2 ≤ i ≤ n.
Choose any x such that 1 ≤ x ≤ arr[i]
Set arr[i-1] to arr[i-1] + x
Set arr[i] to arr[i] - x
Minimize the maximum value of arr using the operation and return the value.
Example
n = 4
arr = [1, 5, 7, 6]
Assuming 1-based indexing.
One optimal sequences is:
Operation 1: choose i = 3, x = 4 (note that x <= arr[3], i.e. 4 < 7).
Replace arr[i-1] with arr[i-1]+x or 5 + 4 = 9
Replace arr[i] with arr[i] - x or 7 - 4 = 3
The array is now [1, 9, 3, 6] (maximum = 9)
Operation 2: i = 2, x = 4
Replace arr[2-1] with 1 + 4 = 5
Replace arr[2] with 9 - 4 = 5
The array is now [5, 5, 3, 6] (maximum = 6)
Operation 3: i = 4, x = 1, the resulting array is [5, 5, 4, 5] (maximum = 5)
The minimum possible value of max(arr) is 5 after operation 3.
Language
C++20
Test
Custom
Type here to search


hackerrank.com/test/6ai3krsls1q/questions/e078e1ja6bl
7m left
The minimum possible value of max(arr) is 5 after operation 3.
Function Description
Complete the function getMaximum in the editor below.
getMaximum has the following parameter:
int arr[n]: an array of integers
Returns
int: the minimum maximum value possible
Constraints
1 ≤ n ≤ 10^5
1 ≤ arr[i] ≤ 10^9
Input Format For Custom Testing
Sample Case 0
Sample Input For Custom Testing
STDIN FUNCTION
3 → n = 3
5 → arr = [5, 15, 19]
15
19
Sample Output
13
Language
C++20
Test


7m left
Constraints
1 ≤ n ≤ 10^5
1 ≤ arr[i] ≤ 10^9
Input Format For Custom Testing
Sample Case 0
Sample Input For Custom Testing
STDIN FUNCTION
3 → n = 3
5 → arr = [5, 15, 19]
15
19
Sample Output
13
Explanation
One sequence that produces the optimal answer is:
Operation 1: i = 3, x = 6 -> [5, 21, 13].
Operation 2: i = 2, x = 8 -> [13, 13, 13].
Sample Case 1
Language
C++20
Test