Codeforces Review

Contents

.1 2013D - 1900

Problem 1. [1] *Given an array A and an operator p that for each a_i and a_{i+1} where $i \geq 1$, p may decrease $|a_i - a_{i+1}|$. p can be performed by infinite times and find the minimum $|max_{i=1}^{n}(a_i) - min_{i=1}^{n}(a_i)|$*

Solution 1. *Suppose b_i is the result of performing p on a_i given the prefix of array A $\{a_1, a_2, \dots, a_i\}$. Create a stack S to load these b_i and the count of it c_i. Then $\{a_1, a_2, \dots, a_i\}$ is stored in S as $(b_1, c_1), (b_2, c_2), \dots, (b_m, c_m)$. And we keep the pairs in ascending order of b_i. For each new a_{i+1}, merge it to the top if $a_{i+1} < b_m$ and then merge the top downwards until $b_k > b_{k-1}$. This way, each a_i is loaded in the array for once, the time of merging until a_i is at most i, so the time complexity if $O(n)$.*

.2 2014H - 1900

Takeaway 1. [2] *A quick way to check if each a_i in array A's slice $A[l:r]$ appears even time:*

1. hash A into a much larger space, for instance, 1 2^{64}. 2. get xor sum of the hash values for the elements in $A[l:r]$. 3. if the xor sum is zero, then each a_i appears an even number of times in $A[l:r]$.

The reason for hashing is to decrease the possibility of mistakes (such that the xor sum is 0 while some of a_i appears odd times, e.g., $\{1, 2, 3\}$), ensuring that the xor sum accurately reflects the even occurrence of elements. The possibility of mistake is $\frac{1}{2^{64}}$.

[1] https://codeforces.com/problemset/problem/2013/D

[2] https://codeforces.com/problemset/problem/2014/H

.3 2035D - 1800

Problem 2. [3] *Given an array A and an operator p that for each a_i and a_j that $i < j$, p can update a_i to $a_i \gg 1$ and a_j to $a_j \ll 1$. p can be performed by infinite times and find the maximum sum of all the prefixes of A.*

Solution 2. *A little similar to 2013D. In 2013D, we store b_i and c_i, the information of a_i after p in a stack S. This approach is appliable in this problem too. In this problem $b_i = min\{b_i^j | b_i^j \ll c = a_i\}$, and c_i is the sum of all c_k that $k < i$ that can reach the maximum prefix sum from 1 to i. This sum is obtained by merging from the top of the stack downwards. If $c_k(k < i)$ is added to c_i, then b_k is popped out from the stack and added to the final sum. After the merging is terminated, we push (b_i, c_i) onto S. Each (b_i, c_i) pair is at most pushed to S by once and popped by once. So the time complexity is $O(n)$*

.4 2009G1 - 1900

Problem 3. [4] *Given an array A and an operator op that can change any element a_i to any another value. Find the minimum step required to make A a consecutive array in which $a_{i+1} - a_i = 1$ for any i.*

Solution 3. *Create an array B that $b_i = a_i - i$. Count the frequency f_i of each b_i and the answer is $A.size() - max(f_i)$.*

.5 2002D1, 2021C2 - 1900

Problem 4. [5][6] *Continuously query about a property P with an update before each query.*

[3] https://codeforces.com/problemset/problem/2035/D

[4] https://codeforces.com/problemset/problem/2009/G1

[5] https://codeforces.com/problemset/problem/2002/D1

[6] https://codeforces.com/problemset/problem/2021/C2

Solution 4. *Find another easy-to-maintain property P' s.t.if P' is satisfied then P is satisfied. The time complexity of checking for P' after each update is $O(1)$ or $O(logn)$.*

.6 1991E - 1900

Takeaway 2. [7] *If I want to use two colors to color an undirected graph G, in which for every v_1, v_2 connected by edge E, their colors are different, the only thing to judge if I can make the coloring is check if the graph is bipartite.*

.7 1991D - 1900

Takeaway 3. [8] *If $a - b = kx$, and x is the power of 2, then $a \bigoplus b = x$ where $\bigoplus$ means XOR*

.8 2039C2 - 1800

Takeaway 4. [9] *Some basic math that is useful for this kind of CF problems. Suppose $a >= b$, then $a - b <= a \bigoplus b <= a + b <= 2a$ since $\bigoplus$ is just addition without carry or subtraction without borrowing. Suppose $a > b$, then $lcm(a, b) <= 2max(a, b) < 2a$.*

[7] https://codeforces.com/problemset/problem/1991/E

[8] https://codeforces.com/problemset/problem/1991/D

[9] https://codeforces.com/problemset/problem/2039/C2