


[questions](#) [tags](#) [users](#) [badges](#) [unanswered](#) | [ask a question](#) [about](#) [faq](#)

CodeChef Discussion

☒ questions ☐ tags ☐ users

UPDTREE - editorial

- PROBLEM LINK:**
- 6 [Practice](#)
[Contest](#)
Author: Lalit Kundu
Tester: Tasnim Imran Sunny
 1 **Editorialist:** Devendra Agarwal

DIFFICULTY:
 Medium-Hard

PREREQUISITES:
 DFS and LCA

PROBLEM:
 You have a tree consisting of N vertices numbered 1 to N.
 Initially each edge has a value equal to zero. You have to first perform M1 operations and then answer M2 queries.
 Operations are defined by:
 A B C D: On the path between nodes numbered A and B increase the value of each edge by 1, except for those edges which occur on the path between C and D.
 Queries are of the following type:
 E F: Print the sum of values of all the edges on the path between two distinct nodes E and F.

Solution:

For solving this problem, we will be solving some sub problems to finally solve this one.

Sub Problem 1

Given 1D array, you need to maintain the array with Update operations of adding X to all index from A to B followed by Query operations of the sum of any subarray.

Solution to Sub Problem 1

Let the array be denoted by Arr[]
 For every Update, do $Arr[B+1] -= X$, $Arr[A] += X$.
 After all Updates, store prefix sum of the Arr[i] at Prefix[i]. It can easily be observed that Prefix Array contains correct numbers at each index.
 Store Prefix sum of Prefix Array in Sum[] array.
 Now for reporting the sum of array elements from x to y: return $Sum[y] - Sum[x-1]$

Sub Problem 2

Given 1D array, you need to maintain the array with Update operations of adding 1 to all index from A to B but leaving indexes which are a part of C to D followed by Query operations of the sum of the subarray.

Solution to Sub Problem 2

Essentially the idea here is we would do +X to all elements from A to B and -X to all elements which are in the intersection of the two segments.

We only aim here at finding intersection of two segments. Rest is managed by the logic of problem 1.

Case 1 : If C is more than B then no intersection.

Case 2 : If A is more than D then no intersection.

Case 3 : If A is more than C then common intersection segment is (A, min(B,D))

Case 4 : If C is more than A then common intersection segment is (C, min(B,D))

Last 2 cases can be merged into 1 as (max(A,C), min(B,D)) is the common intersection if intersection happens.

Removing Cases 1 and 2: if (max(A,C) > min(B,D)) then no Intersection

Sub Problem 3

Given a tree, you need to maintain the tree with Update operations of adding X to all Edges in the path from A to B followed by Query operations of the sum of all Edges in the path from E to F.

Solution to Sub Problem 3

Each edge can be mapped to a vertex in the tree.

Can you find the mapping ?

Fix the root. Now consider the edge to be between v1 and v2. Associate the edge to v1 if v2 is the parent of v1 or vice versa.

Follow this question

By Email:

You are not subscribed to this question.

(you can adjust your notification settings on your [profile](#))

By RSS:

Answers

Answers and Comments

Tags:

[editorial](#) ×7,142

[dp](#) ×1,145

[medium-hard](#) ×501

[dfs](#) ×272

[cook50](#) ×114

[lca](#) ×91

Asked: 22 Sep '14, 00:14

Seen: 3,969 times

Last updated: 01 Oct '14, 21:23

Related questions

[MAZETRAV](#) - Editorial

[PALSTR](#) - Editorial

[MGCHGYM](#) - Editorial

[ICLFIN05](#) - Editorial

[CHN15F](#) - Editorial

[THEGAME](#) - Editorial

[CHKTRE](#) - Editorial

[TREEP](#) - Editorial

[SUBLCM](#) - editorial

[CLDSIN](#) - Editorial

Path Update from A to B : Let anc be the ancestor of A and B. Add X to Arr[A] and -X to Arr[anc]. Add X to Arr[B] and -X to Arr[anc].

After all update queries , run dfs and $\text{Prefix}[\text{node}] = \text{Arr}[\text{node}] + \text{Prefix}[\text{all_childrens}]$

Run another Dfs which calculates the sum of all nodes in the path from root to that node and store in that node. Let us denote this array as Sum[]

Report Query: Let anc be the ancestor of E and F. Return $\text{Sum}[E] + \text{Sum}[F] - 2 * \text{Sum}[\text{anc}]$

Essentially the idea is that the leaf node is the point our array starts and each path from leaf to root is a 1D array

Subproblem because each node has a unique parent.

Are you tired of reading ? Next we come up with the main problem and you will solve it and i will just question you !! Are you ready ?

Main Problem

Refer to Problem above :)

Solution

Given Intersected Segment , can you correctly update the tree ? [Hint : Sub Problem 2]

Idea for handling intersection is same as Sub problem 2 i.e. we add +X to all nodes in the path from A to B and -X to all nodes in the common Intersection.

We only aim at finding the intersection of two paths , rest is handled by Sub-problem 3.

Let anc1 be the ancestor of A and B. Let anc2 be the ancestor of C and D.

We will deal with all four pairs of disjoint Paths separately i.e [A anc1 C anc2] , [A anc1 D anc2] , [B anc1 C anc2] and [B anc1 D anc2].

Let's us see how to solve any one of them , rest is just a function calling :)

Let us solve [A anc1 C anc2] (Unbiased Choice :))

Can you find it using sub problem 2?

anc2 must lie in the path from A to anc1 otherwise there will be no intersection [Reasoning: If there is a intersection and anc2 is not in their path then the intersection must be of a Cross Format but this is not possible because root --> anc1 , root --> anc2 , anc1 --> Cross_Vertex , anc2--> Cross_Vertex form a cycle and we know that in tree , there are no cycles.]

If anc2 is not the ancestor of A and anc2 , then there is no intersection. Reason : Follows from above. Let x be the ancestor of A and C. Then the problem reduces to finding intersection in 1D , given two segments [A anc1] and [x anc2].

Can you solve this ?

$\max(A,C)$ is lca(A,x) , $\min(B,D)$ is other vertex which is not lca(x,B).

Now you can see intersection is only possible if $\text{lca}(\max(A,C) , \text{lca}(A,x))$ is $\min(B,D)$

Extra Input at the end

We missed to check in our test data about the extra input than required. Thanks to [lyrically](#) for pointing this out. We hope that such error will not happen in future. We are extremely sorry for this :(.

Time Complexity

$O(\log n)$ per update Query.

$O(N)$ preprocessing.

$O(\log n)$ per Query [Time required to find ancestor].

AUTHOR'S and TESTER'S SOLUTIONS:

Author's solution

Tester's solution

[editorial](#) [dfs](#) [cook50](#) [medium-hard](#) [lca](#) [dp](#)

This question is marked "community wiki".

asked 22 Sep '14, 00:14

devuy11 ♦♦
[46] • 27 • 38 • 42
accept rate: 0%

edited 01 Oct '14, 21:23

So, let's get to bugfixes.

- "Contest" link is wrong
- Author's and tester's solution give Access Denied

[xellos0](#) (22 Sep '14, 00:18)

upd: now the tester's solution is fixed, the other bugs are still there

[xellos0](#) (22 Sep '14, 03:20)

upd2: now just the contest link is wrong.

[xellos0](#) (24 Sep '14, 10:50)

now all links are fine :)

[dpraveen](#) ♦♦ (24 Sep '14, 13:08)

2 Answers:

oldest newest most voted

6 Devendra Agarwal writes wonderful editorials. Especially for (medium-hard and hard) TREE related problems. I have also enjoyed reading his editorials at hackerrank. Great Work ! :-)

[link](#) | [award points](#)

edited 23 Sep '14, 02:23

answered 23 Sep '14, 02:19



rsaha77

[965] ●3●8●15

accept rate: 17%

0 "anc2 must lie in the path from A to anc1" shouldn't this be "anc2 must lie in the path from A to ROOT"? @admin Please clarify.

[link](#) | [award points](#)

answered 23 Sep '14, 00:45



flaminrage

[244] ●2●3●11

accept rate: 0%

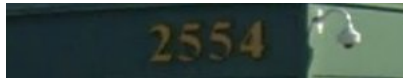
1 No , it should be anc1 only because if anc1 is in between anc1 and Root then there is no common intersection as ancestor of anc1 and A is anc2 [which implies that there is no common node in between anc1 and A]

[devuy11](#) ♦♦ (24 Sep '14, 13:15)

Your answer

[hide preview]

☐ community wiki



Type the text

[Privacy & Terms](#)



Post Your Answer

[About CodeChef](#) | [About Directi](#) | [CEO's Corner](#)
[CodeChef Campus Chapters](#) | [CodeChef For Schools](#) | [Contact Us](#)

© 2009, Directi Group. All Rights Reserved.
 Powered by OSQA

Directi
 Intelligent People. Innovative Ideas.