Beauty of tree (13pts, 19pts)

Practice Submissions

You have not attempted this problem.

Last updated: Jul 14 2020, 21:09

PROBLEM

ANALYSIS

Analysis

For a given node, the P(visited by either Amadea or Bilva) = 1 - P(not visited by Amadea nor Bilva).
This leads to P(visited by either Amadea or Bilva) = 1 - ((1-P(visited by Amadea)) * (1-P(visited by Bilva)))
i.e. For a given node, note that events of visited-by-Amadea and visited-by-Bilva are mutually independent events, and hence P(being visited by either Amadea or Bilva) = P(visited by Amadea) + P(visited by Bilva) - (P(visited by Amadea)*P(visited by Bilva))

Given the above formula, our goal now is to find out P(visited by Amadea) and P(visited by Bilva) for every node in the tree. We can use DFS in order to do this.
Firstly, let's define 2 variables visits_a[] and visits_b[] where visits_a[i] and visits_b[i] denote the number of visits to node-i across all paths starting from any node in the subtree of node i with skips **A** and **B** respectively.

Now, the first DFS run would be from node-1 to compute visits_a[]. As we perform this DFS, we can keep a track of the path that has been taken so far, let's say path_taken[]. As we enter a node-i, we add it to path_taken[] and call DFS on it's children. Once we come back to node-i, we remove it from the path_taken[] and increment visit_a[i] by 1. Note that this increment is for the path leading from node-i to itself. Next, we check if there is some node-**j** which is **A** skips behind node-i. Using path_taken[], we check to see if such a node is present and increment the visits_a[node-j] by visits_a[i]. At the end of this DFS run, we have the total visit count for all nodes with the skip distance of **A**. Now, dividing visits_a[] by **N** gives us the P(visited by Amadea) for each node in the tree. Repeat the above process for to compute visits_b[] and obtain P(visited by Bilva) for every node in the tree.

With P(visited by Amadea) and P(visited by Bilva) computed for every node in the tree, computing and summing over P(being visited by either Amadea or Bilva) for each node will give us the answer. Since DFS takes linear time in the number of vertices, the time complexity of the solution is O(**N**).