

## Question 4

Let's take as a graph  $G$  linear directed tree (path) of the length  $n$ . Suppose the algorithm  $\mathcal{A}$  finds the set of vertices from the problem task in less than  $\Omega(\log^* n)$ . Then we will show that extending it with steps which cost only  $O(1)$  could achieve 3-coloring of  $G$ . This will be in contradiction with the Theorem 1.7 from lecture book [1], which says that any algorithm that does this coloring on a directed path should be at least  $\frac{\log^* n}{2} - 2$ .

So, as an extension of  $\mathcal{A}$  we propose that after it finishes **all marked nodes** take color 1 and send message about this to neighbors, then stop.

**Any other node** does this in 3 cases of receiving messages:  
if receive one message about the color  $i$  of its neighbor, it takes the color  $3 - i$  and send message about its color to its other neighbor and stops;  
if this node receives message already after stopping, it compares its ID with ID of the neighbor who sent this message (suppose when sending color they send their ID too) and colors itself to the color 3 if its ID is less;  
if node receives two messages (can't receive more in path, so all cases are considered here) it takes the color 3 and stops (in such a case its neighbors can't have color 3, as they propagated signal, and this node doesn't).

In such a way propagation waves will go from marked nodes until they reach the end of tree (then all the nodes in-between are colored properly in alternating 2-1-...) or collide to each other in the middle of "pure" path and color one of the nodes in the middle to 3 still having proper coloring.

As any node of the graph should be at most at the distance 100 from some marked initially nodes, therefore "pure" parts (chains) of path (without marked nodes) are no more than 200 in length (radius no more than 100) and therefore extension of  $\mathcal{A}$  will take no more than  $1 + 100 = 101 = O(1)$  rounds. Therefore overall we still will have less than  $\Omega(\log^* n)$ .

But we also have proper 3-coloring, as marked nodes have color 1 and can't be neighbors to each other from the task description. Then, propagation makes every next node in "pure" paths to have color different from its neighbors (if there is even number of nodes in "pure" path, they will be colored to alternating 2-1-..., but in the middle instead of two 2s one of them with smallest ID will have color 3; if number of nodes in "pure" path is odd, then just would be alternating 2-1-... with 3 in the middle) and all colors are from 1 to 3.

Therefore we conclude the contradiction and we proved the claim.

## References

- [1] M. Ghaffari, "Distributed graph algorithms." <https://disco.ethz.ch/courses/fs19/podc/lecturenotes/LOCAL.pdf>.