{121, 112}

Upper bound

Always use 1->1->1->... for "larger" subtree and 1->2->1->2... for "smaller" subtree, so that the alternating paths 1->2->1->2... that you create can be as short as possible. E.g. if you have a long path with some short hairs, the long path would be 1->1->1->... and only the hairs would need color 2 in a coordinated manner.

Hence $O(log\ n)$

Lower bound

It is clear that, in a general case, we cannot use only 1 of the 2 labels, so both labels will need to be used. Therefore, trivial solution where each node outputs the same label wouldn't work here.

It is rather easy to see that a deterministic setting there is no fast algorithm since, looking at a regular tree, at some level x, labels of the nodes at this level will **uniquely** determine labels of their parents => we have $\Omega(logn)$.

We can argue that randomness does not help here. Let's assume the opposite. Then there is a node v somewhere in the middle of a large rooted tree G s.t. the output of v depends on the random bits in its constant neighbourhood C. In particular, the output of v is not completely determined by unique ids of nodes in C and the tree structure. Then, if we look at the set X of nodes s.t. they are at the distance of d from v and all nodes in X are descendents of v. Given that d is large enough, the number of label-2 nodes in X has to change whenever v changes its label. Thus, no matter what's the distribution of labels in set X, if their random bits are independent of the random bits of v, with probability $> p$ we will have a mismatch somewhere in the tree.

Hence $\Omega(log\ n)$

Thus, we have $\Theta(log\ n)$