Easter Eggs

-editorial-

Author: Oncescu Costin
Preparation: Niculae Alexandru

The Public Garden is a tree with N vertices

N queries:

You do a query on each island.

N - 1 queries:

You do a query on each island from 1 to $N-1$. If the egg wasn't found, it must be on the N-th island.

N - 2 queries:

You do a query on two connected islands. If the egg is among them, with one more query you will get the answer. If not, then you should proceed as in the previous subtask: You have $N-2$ islands remained, and you can get the answer with $N-3$ more queries. Total queries: $max(1+1, 1+N-3)$.

$\frac{\boldsymbol{Log_2{}^2N}}{\mathbf{2}}$ ***queries:***

If we have some fixed root, let sub_{node}- the vertices from node's subtree, inclusive node. At each step, compute the centroid of the tree. Let X be the current centroid, and son_1, son_2, ...son_k, the sons of X (X has k sons). If we consider the following array of sets: $\{ X \}$, $\{sub_{son_1}\}$, $\{sub_{son_2}\}$, … $\{sub_{son_k}\}$, it can be easily observed that every prefix of this array is connected (the subtrees are connected by X).

Doing a binary search on prefixes, with $log_2(K+1)$ queries you can find where the egg is. If the egg is in X, you print the answer and stop. Otherwise, you go in the subtree where the egg is, and do the same thing from the previous step.

$(\boldsymbol{Log_2N})+\mathbf{1}$ ***queries:***

It can be observed that every segment from the Euler Tour is connex, so there can be done a binary search on the Euler Tour. (Euler Tour has length $2 * N-1$).

$\boldsymbol{Log_2\, N}$ ***queries:***

Every prefix of dfs/bfs is connex, so it can be done a binary search on prefixes of dfs/bfs and find where the egg is. You should pay attention to the implementation, as there will be obtained $(\log_2 N)+1$ queries if the egg is the last in the dfs/bfs array and N is a power of 2. You have to consider only the first $N-1$ islands in the binary search, and if the egg wasn't found, it must be in the last island.

This is also optimal: If we do X queries and look at the queries answers as they are some bits in base 2, we will get a binary representation of length X (let's call it S). If we don't have an algorithm based on random (and we don't have), we will get a different S every time the egg is hid in a different place. As for X queries, you can get 2^X different S representations (because S has length X), $\log_2 \mathrm{N}$ queries is the best you can get for N vertices, as $2^{\log_2 N}=N$.