# Puma – Analysis

Problem: Iliyan Yordanov, Analysis: Encho Mishinev

The input structure is a tree. We will refer to places as nodes, with place 0 being the root of the tree. The exits are the leaves of the tree. Since all nodes must be visited and we have as many zoologists as leaves, it is necessary and sufficient to make sure that each zoologist exits from a different leaf node.

## Subtask 1

Node 0 is directly connected to all other nodes. Since the input is a tree, there are no other edges and we have a star graph centered at 0.

The strategy is straightforward – mark each leaf node with 1 before exiting, then when starting in the root just go to any leaf that is still marked with 0. This uses $M = 1$ as required,

## Subtask 2 – solution with $M = N - 1$

There are several different solutions using markings up to $N$. Here we will describe one such that we will later adapt for full score.

Let us reserve the marking of 0 to indicate a node has never been visited. For a visited node $v$, we would like to maintain the marking as a value $F(v)$ that indicates how many unvisited leaves there are in the subtree of $v$. In order to keep the meaning of 0 separate, we will store this value offset (i.e. increased) by 1. For example, storing a marking 5 would mean that there are 4 unvisited leaves in this node's subtree.

We choose our paths as follows:

1. If there is any child which is still marked with 0, we go there, since that node has never been visited and must lead to at least one unvisited leaf.
2. Otherwise, we find any child that has a marking of 2 or more and go there, since that indicates there are 1 or more unvisited nodes in its subtree.

Let us see how to update the markings in order to maintain these values correctly. Let the total number of leaves in the subtree of node $v$ be denoted by $S(v)$. We can compute all $S(v)$ values in $O(N)$ by a simple DFS traversal of the tree.

Then we choose the markings as follows:

1. If the current marking is 0, then this is the first time we are visiting this node. It currently has $S(v)$ unvisited leaves, but we will visit one of them, so $S(v) - 1$ will remain after we exit. Since our notation is offset by 1, we set $F(v) = (S(v) - 1) + 1$.
2. If the current marking is not 0, then we have visited this node in the past and its current value is correct. Since we will reduce the number of unvisited leaves by passing through, we set $F(v) = F(v) - 1$.

This scheme guarantees that every zoologist will exit from a different leaf. Since $S(v) \leq N - 1$, we use only markings up to $N - 1$.

## Full Solution

We will extend the previous solution to only use markings up to $M = C$, where $C$ is the length of the longest vertical path in the tree.

We will have the exact same ruleset for moving, and we will adapt only the marking rules. The idea will be to store $M$ instead of the exact value whenever the number of leaves in the node's subtree exceeds $M - 1$. Hence $M$ would mean "some large amount of unvisited leaves, yet unknown". The exact rules are as follows:

1. If the current marking is 0, then this is the first time we are visiting this node. As explained earlier, the correct value is $S(v)$, but we set $F(v) = \min(S(v), M)$.
2. If the current marking is not 0, but there is at least one child with marking 0, then we are still during the "first visits" for children. Let there be exactly $T$ children with markings different from 0. Since we always go towards a 0-marked child if there is one, we know that we have been in this node exactly $T$ times before. Taking into account the current visit, the correct marking is $(S(v) - T - 1) + 1$. As noted, we set $F(v) = \min(S(v) - T, M)$.
3. Otherwise, if no children have marking $M$, we update the current node's marking using its children. The number of leaves in its subtree according to its children's marking would be: $W = \sum_{child} (F(child) - 1)$. The minus one is just due to the storing offset. Taking the current visit into account, the correct offset marking would be $(W - 1) + 1$, so we set $F(v) = \min(S(v), W)$.
4. Otherwise, if there is at least one child with a marking of $M$, we set $F(v) = M$. This is the only problematic case, since this can break the exact meaning of the marking. We can try to avoid breaking it, by always prioritizing children with marking $M$ when moving. If we do that, our markings have their intended meaning, as long as no node has only one child (which is a subtask).

It might not be immediately obvious why this scheme works, or why it needs $M = C$, so let us explore it a bit more in-depth.

Since a marking of $M$ has a special meaning, let us call nodes having a marking of $M$ *heavy*. We will also call a marking *correct* if it reflects the correct number of unvisited leaves in the node's subtree at that moment.

For a node $v$, if $S(v) < M$, then neither it nor any node in its subtree will ever be heavy and hence will not be affected by the min-capping. Thus their markings remains correct at all points, just like the solution in the previous subtask.

Let us then consider heavy nodes and specifically consider the moment they stop being heavy. Let us call the moment node $v$'s marking changes from $M$ to something else $v$'s *correction*, since before that the marking might not be *correct*.

We make the following claim:

**Claim.** *Immediately after $v$'s correction, its new marking will be correct and will be* ***at least*** *$M - d$ where $d$ is the length of the longest vertical path in $v$'s subtree.*

We prove this inductively on $d$ (the length of longest vertical path in a node's subtree):

- Base case $d = 1$. These are nodes which have only leaves as children. Upon visiting such a node we can always compute the correct marking, but write $M$ if it is too big. Hence, at the *correction* of $v$ its marking will be precisely $M - 1$, since the correct markings can only reduce by 1 per visit.
- Induction step $d > 1$. Consider $v$'s *correction* moment.
  - If upon the last visit of $v$ it still had unvisited children, or did not have any heavy children, then we were able to compute the *correct* marking and wrote $M$. Since the number of unvisited leaves can reduce by at most 1 per visit, then the marking of $v$ must now be being set to the *correct* $M - 1$
  - Otherwise, there was a child $c$ of $v$ that was heavy the last time we visited $v$, but is not heavy now (as otherwise this wouldn't be $v$'s *correction*). This means that $c$'s *correction* happened the last time we moved from $v$ to $c$, and $c$ has not been visited since. By induction, $F(c)$ must have been set to at least $M - (d - 1) = M - d + 1$,

since the longest vertical path in $c$'s subtree is at most $d-1$. It follows that $v$'s *correction* will set $F(v) \geq M-d$, since it uses the sum of the child markings (see case 4 above). Furthermore, by induction all children have *correct* markings and so the computed marking for $v$ is also *correct*.

If $C$ is the longest vertical path from the root, then any $v$ that is not the root will be marked as $F(v) > M - C$ upon its correction. For $M = C$ we get $F(v) > 1$, which means the solution will work correctly (note that the *correct* marking takes into account the current zoologist). If we were to try and use $M < C$, then by this scheme a node's marking might change from $M$ to $0$, which would mean the current zoologist cannot reach an unvisited leaf.

It is possible to construct similar schemes with slightly different interpretation of the markings, but informally, they all take $C$ "iterations" to propagate information up the tree, since a single zoologist can essentially only "move" information up one level.

Note: Make sure that you handle $N = 1$ correctly, as that is a valid input for the final subtask.