Principles of Distributed Computing
Exercise 7: Sample Solution

1 Concurrent Ivy

a) The three nodes are served in the order \( v_2, v_3, v_1 \).

b) Figure 1 depicts the structure of the tree after the requests have been served. Since \( v_1 \) is served last, it is the holder of the token at the end.

![Figure 1: Tree after the requests have been served.](image)

2 Tight Ivy

a) In order to show that the bound of \( \log n \) steps on average is tight, we construct a special tree which is defined recursively as follows. The tree \( T_0 \) consists of a single node. The tree \( T_i \) consists of a root together with \( i \) subtrees, which are \( T_0, \ldots, T_{i-1} \), rooted at the \( i \) children of the root, see Figure 2.

First, we will show that the number of nodes in the tree \( T_i \) is \( 2^i \). This obviously holds for \( T_0 \). The induction hypothesis is that it holds for all \( T_0, \ldots, T_{i-1} \). It follows that the number of nodes of \( T_i \) is \( n = 1 + \sum_{j=0}^{i-1} 2^j = 2^i \).

We will show now that the radius of the root of \( T_i \) is \( R(T_i) = i \). Again, this is trivially true for \( T_0 \). It is easy to see that \( R(T_i) = 1 + R(T_{i-1}) \), because \( T_{i-1} \) is the child with the largest radius. Inductively, it follows that \( R(T_i) = i \).

By definition, when cutting off the subtree \( T_{i-1} \) from \( T_i \), the resulting tree is again \( T_{i-1} \). Let \( C : T_i \mapsto T_{i-1} \) denote this cutting operation. For all \( i > 0 \), we thus have that \( C(T_i) = T_{i-1} \).
We will now start a request at the single node $v$ with a distance of $i$ from the root in $T_i$. On its path to the root, the request passes nodes that are roots of the trees $T_1, \ldots, T_i$. All of those nodes become children of the new root $v$ according to the Ivy protocol. The new children lose their largest “child” subtree in the process, thus the children of node $v$ have the structures $C(T_1), \ldots, C(T_i) = T_0, \ldots, T_{i-1}$. Hence, the structure of the tree does not change due to the request and all subsequent requests can also cost $i$ steps. Since $n = 2^i$, each request costs exactly $\log n$.

b) The access pattern we described above already has the property that each node requests the object in sequence. We can show this inductively over $i$ for the trees $T_i$. First we introduce some additional notation. We consider a tree $T_i$, for any $i > 0$, as two parts: The left subtree $L(T_i)$, which has the structure of $T_{i-1}$, and the rest of the tree $R(T_i)$, which also has the same structure has $T_{i-1}$. We then write $T_i = L(T_i) \rightarrow R(T_i)$ to indicate that $T_i$ is the tree obtained by rooting $L(T_i)$ as the left-most child of the root in $R(T_i)$. We note that with this notation one iteration of Ivy handling a request from the highest-depth leaf performs a tree rotation that can be described recursively as $\text{Rot}(T_i) = \text{Rot}(R(T_i)) \rightarrow L(T_i)$. We further write $\text{Rot}^k(T_i) = \text{Rot}(\text{Rot}^{k-1}(T_i))$ with $\text{Rot}^0(T_i) = T_i$.

We can now show this inductively over $i$ for the trees $T_i$. We will start with $T_1$ as the base case since our notation only works for $i > 0$ and the case for $T_0$ is trivial. In the first iteration on $T_1$ the leaf node requests the object, after that the edge is switched and the previous root node requests the object. For the inductive step we observe that over $i$ iterations of the access pattern above Ivy accesses the highest-depth leaves of the trees $T_i, \text{Rot}(T_i), \ldots, \text{Rot}^{2^i-1}(T_i)$. Unwinding the definition of Rot we see that these correspond to the highest-depth leaves of $L(T_i), \text{Rot}(L(T_i)), \ldots, \text{Rot}^{2^i-1}(L(T_i))$ on even (zero-indexed) iterations, and $\text{Rot}(R(T_i)), \text{Rot}^2(R(T_i)), \ldots, \text{Rot}^{2^i-1}(R(T_i))$ on odd (zero-indexed) iterations. According to the inductive hypothesis these iterate through the $2^i-1$ nodes of the two subtrees $L(T_i)$ and $R(T_i)$. Thereby the alternation of the two iterates over all $2^i$ nodes of $T_i$. 

![Figure 2: The trees $T_0, \ldots, T_3$.](image)