Albert-Ludwigs-Universität
Institut für Informatik
Prof. Dr. F. Kuhn
Mohamad Ahmadi, Hamid Ghodselahi

June 01, 2016

Network Algorithms, Summer Term 2016
Problem Set 4 – Sample Solution

Exercise 1: Concurrent Ivy

1. The three nodes are served in the order v_2, v_3, v_1.
2. 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.

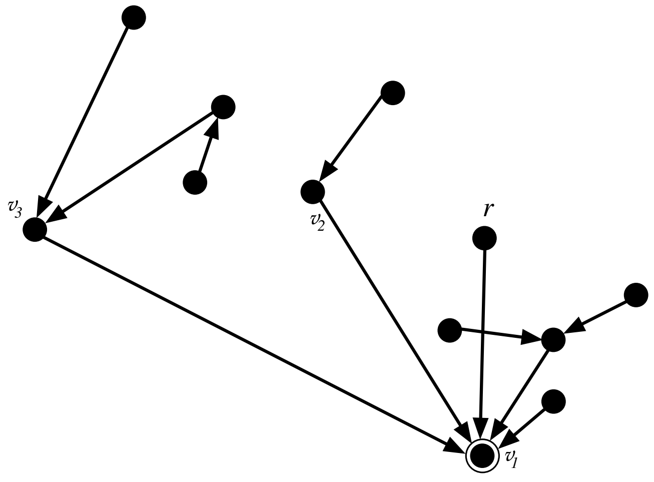

Figure 1: Tree after the requests have been served.

Exercise 2: Tight Ivy

In order to show that the bound of $\log n$ steps on average is tight, we construct a special tree, called *Binomial Tree*, which is defined recursively as follows. The tree $\mathcal{T}_0$ consists of a single node. The tree $\mathcal{T}_i$ consists of a root together with i subtrees, which are $\mathcal{T}_0, \ldots, \mathcal{T}_{i-1}$, rooted at the i children of the root, see Figure 2.
First, we will show that the number of nodes in the tree $\mathcal{T}_i$ is 2^i. This obviously holds for $\mathcal{T}_0$. The induction hypothesis is that it holds for all $\mathcal{T}_0, \ldots, \mathcal{T}_{i-1}$. It follows that the number of nodes of $\mathcal{T}_i$ is $n = 1 + \sum_{j=0}^{i-1} 2^j = 2^i$.
We will show now that the radius of the root of $\mathcal{T}_i$ is $\mathcal{R}(\mathcal{T}_i) = i$. Again, this is trivially true for $\mathcal{T}_0$. It is easy to see that $\mathcal{R}(\mathcal{T}_i) = 1 + \mathcal{R}(\mathcal{T}_{i-1})$, because $\mathcal{T}_{i-1}$ is the child with the largest radius. Inductively, it follows that $\mathcal{R}(\mathcal{T}_i) = i$.
By definition, when cutting of the subtree $\mathcal{T}_{i-1}$ from $\mathcal{T}_i$, the resulting tree is again $\mathcal{T}_{i-1}$. Let $\mathcal{C} : \mathcal{T}_i \mapsto \mathcal{T}_{i-1}$ denote this cutting operation. For all $i > 0$, we thus have that $\mathcal{C}(\mathcal{T}_i) = \mathcal{T}_{i-1}$. We will now start a request at the single node v with a distance of i from the root in $\mathcal{T}_i$. On its path to the root, the

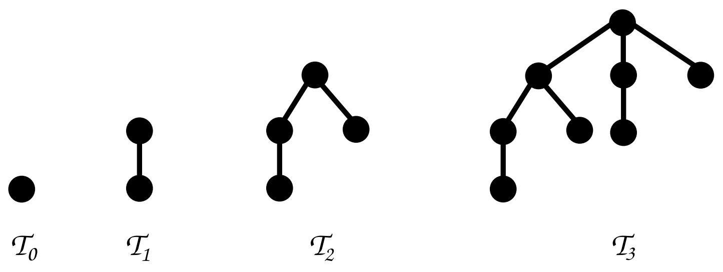

Figure 2: The trees $\mathcal{T}_0, \ldots, \mathcal{T}_3$.

request passes nodes that are roots of the trees $\mathcal{T}_1, \ldots, \mathcal{T}_i$. 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 $\mathcal{C}(\mathcal{T}_1), \ldots, \mathcal{C}(\mathcal{T}_i) = \mathcal{T}_0, \ldots, \mathcal{T}_{i-1}$. 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 $\log n$.