Exercise 4

Preparations for the Proof (Lemma?)

Define the order of a binomial tree as the height of that binomial tree.

Observe the given binomial tree, we can discover that:

1. Binomial tree of order 0 has 1 nodes;
2. Binomial tree of order k is generated by putting two binomial tree of order $k - 1$ together by connecting their root;
3. Combining 1. and 2., we have that binomial tree of order k has exactly 2^k nodes;
4. Root of binomial tree of order k has exactly k children.

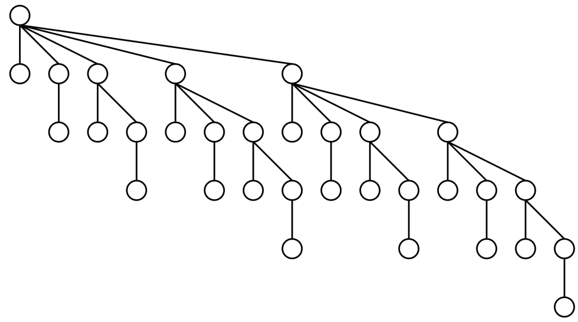

Figure 1: A binomial tree given by the Lecture Note

In the context of broadcast, we know that the total node of the binomial tree is p. It follows that the order of the binomial tree is $k = \log p$ (or we need some rounding up).

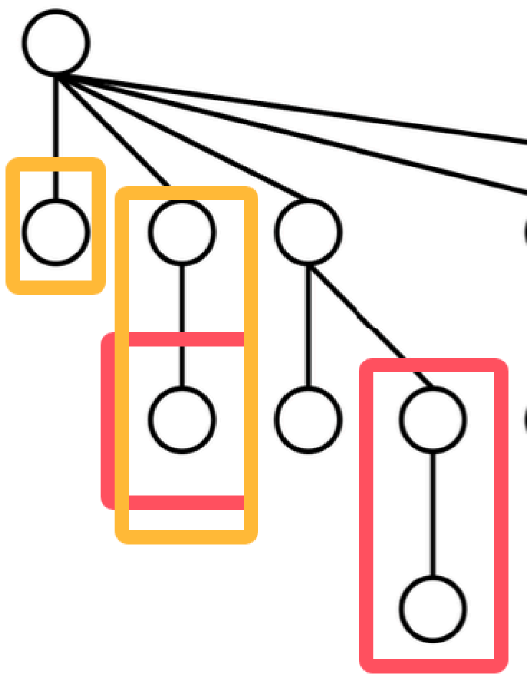

Figure 2: The smaller children is the same with the right children of the bigger children

If the smaller children are preferred, when the nodes start to send messages, as is shown on the Figure 1, we will process the children from left to right. We can note that the cost of processing the smaller children is always smaller than the cost of bigger children (See Figure 2). So the subtask of broadcast for the smaller children will always finish earlier than the bigger children.

The statements above form the basis of our proof.

Proof

- First, it takes **$\log p$ steps** for the root node to inform all of its children of the message;
- Second, we find that the last children informed is of order $\log p - 1$, and it's manifest that it can not be processed in a instant (or $O(1)$). We need another $\log p - 1$ steps for this children to inform their children of the message;
- At last, it took $\log p + \log p - 1 + ... + 1$ steps to make all the PEs informed, which sums up to $\log p(\log p + 1)/2$. This gives the time complexity $\Omega(\log^2 p)$. The Ω-notation arises because of the rounding up of $\log p$.■