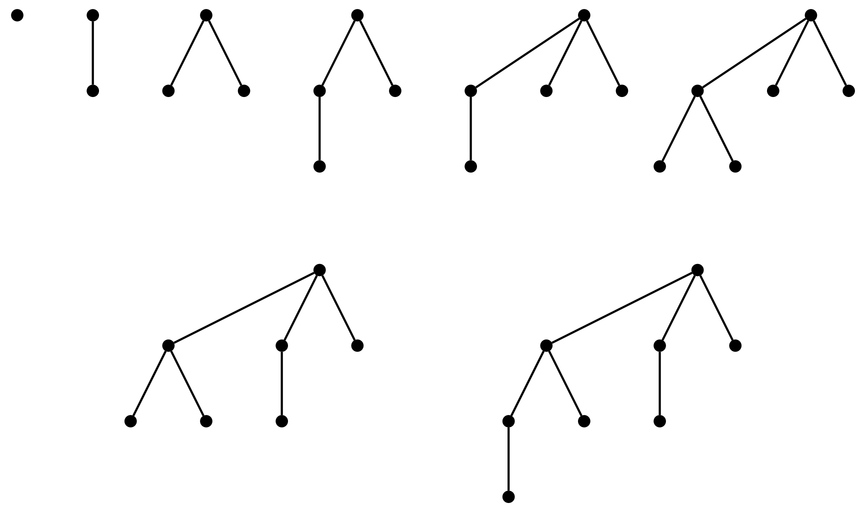
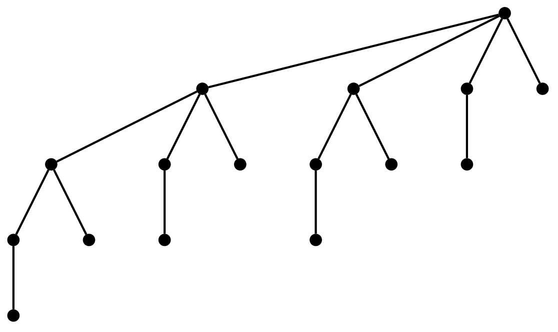
Analysis of Algorithms | **H3 Q3(b) Explanation**
CS6820 Fall 2022 | **Monday, October 17, 2022**

The solution to Homework 3, Question 3(b) may look quite mysterious. How might you have come up with this? Suppose you started with a singleton node, which is the binomial tree B_0. In each step, you merge the current tree into a new singleton and compress the leftmost path. As you go up the path, you see that if you attach shorter subtrees to the new root on the right and longer ones to the left, this tends to keep long paths on the left side. The longest path always seems to be the leftmost one.

You observe that after one step you have B_1, after three steps you have B_2, and after seven steps you have B_3.



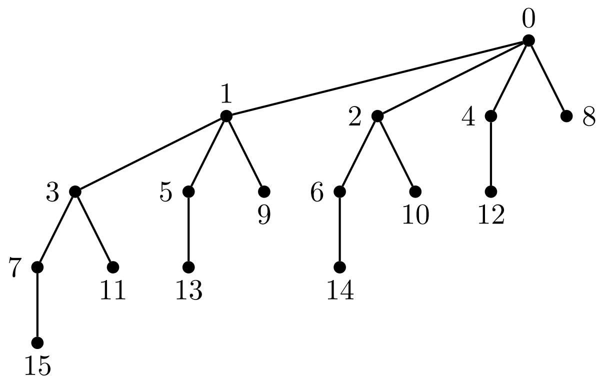
Intrigued by this observation, you keep going. You find that after 15 steps you have B_4.



Now you realize something interesting is going on. You conjecture that after $2^k - 1$ steps you will have B_k. Suddenly you remember the title of the exercise: "Self-reproducing binomial trees." This must be what was meant!

You realize that the leftmost path of B_k is of length k, so you can always choose that one to reduce. You stare at the trees and observe that each tree in succession looks like the last with a new node attached somewhere. Thus each tree you see after B_k contains B_k as a subtree, so must also contain a path of length k. So you get paths of length k after every step until you see B_{k+1}, at which point you get paths of length $k+1$. You think: if you can prove this, it will give the desired lower bound.

You realize that you can prove this if you can find some way to describe the intermediate trees. The B_k are fine, but what about the others? You go back to the observation that each successive tree looks like the previous tree with a new node attached. On a whim, you decide to label the nodes with the times they are attached to see if there is any pattern.



You stare at the numbers and see a definite pattern. You imagine doing this forever and labeling every node of an infinite countably branching tree. You wonder: How do I describe the labeling function? It is a neat puzzle. You get excited and stay up till 4am figuring it out.