COS 488 Exam Question #1

Tim Ratigan

March 16, 2017

Question: We define an *ordered binary tree* to be a binary tree on N nodes where the nodes are labeled and the label of a parent node is always less than each of its roots. How many ordered binary trees of size N are there?

Solution: There are two ways to go about this, one more fruitful than the other. The first is to set up a recursion on the leaves. Since a tree on N nodes has $N+1$ leaves, we can add a node labelled $N+1$ to any of the leaves of an ordered binary tree on N nodes to obtain an ordered binary tree on $N+1$ nodes. If $\mathcal{T}$ is the combinatorial class of ordered binary trees, this gives us that for the EGF,

$$\mathcal{T} = Z \times \mathcal{T} + \epsilon$$
$$T(z) = \frac{1}{1-z}$$

In particular the number of such trees in $N!$ for any N.

The alternative solution is to note that the root node for the whole tree must be labelled 1. Then, recurring on the root, if t_N is the number of ordered binary trees with N nodes then you obtain $t_N = \sum_{i=0}^{N-1} \binom{N-1}{i} t_i t_{N-i-1}$ since you need to choose i of the remaining $N-1$ labels to assign to the left subtree. If one can identify the pattern, then this gives an inductive proof, since $\sum_{i=0}^{N-i} \binom{N-1}{i} i!(N-i-1)! = \sum_{i=0}^{N-1} (N-1)! = N!$