8 Counting binary trees: a detour

8.1 Stirling's approximation

$$\text{Since } \frac{n^n}{n!} \le e^n,$$
$$\left(\frac{n}{e}\right)^n \le n! \le n^n$$

which, surprisingly, gives quite a good estimate. It is not Stirling's approximation. The left-hand side is an underestimate, and the Stirling approximation is

$$n! \sim \sqrt{2\pi n}\left(\frac{n}{e}\right)^n$$

which is asymptotically correct in the sense that the ratio between the left and right sides converges to 1.

8.2 Counting binary trees

Generating functions can be used to compute the number of binary trees with n nodes. Let b_n be the number of binary trees with n nodes:

$$b_0 = 1.$$

Given a tree with n nodes, where $n > 0$, suppose that the left subtree at the root has size i and the right subtree has size j. Then i is the inorder rank of the root, and $i + j = n - 1$. There are b_i possible left subtrees and b_j possible right subtrees. Summing for $i = 0, \ldots, n-1$, we get

$$b_n = \sum_{i+j=n-1} b_i b_j \quad \text{if } n > 0$$

Let $B(z) = \sum_n b_n z^n$, a 'generating function.'

$$\begin{aligned}
B^2(z) &= \sum_i b_i z^i \sum_j b_j z^j = \sum_{n\ge 0}\sum_{i+j=n} b_i b_j z^n \\
&= \sum_{n\ge 1}\sum_{i+j=n-1} b_i b_j z^{n-1} \\
&= \sum_{n\ge 1} b_n z^{n-1} \\
&= (B(z)-1)/z \\
zB^2(z) - B(z) + 1 &= 0
\end{aligned}$$

Now solve this as a quadratic equation for $B(z)$.

$$B(z) = \frac{1 \pm \sqrt{1-4z}}{2z}$$

There are two parts to the solution, one with a positive square root and one with a negative. The positive square root is invalid since it contains a term in $1/z$. This leaves

$$B(z) = \frac{1 - \sqrt{1 - 4z}}{2z}$$

This can be expanded using the binomial theorem and the expression $(1 - 4z)^{1/2}$. The calculation is fairly simple and the answer is

$$b_n = \frac{1}{n+1}\binom{2n}{n}.$$

These are the *Catalan numbers*.

8.3 Average pathlength in binary trees (as opposed to binary search trees)

This part is rather tough, and we shall skip it, except for the important point made in Corollary 8.1.

This is a considerably trickier calculation with a generating function. It comes from Knuth, the art of computer programming, probably volume 1.

Let $b_{n,p}$ denote the number of binary trees with n nodes and internal path length (sum of node depths) p. Let $B(x, y)$ be a bivariate generating function

$$B(x, y) = \sum_{n,p} b_{n,p} x^n y^p$$

$$B(z, 1) = \sum_{n,p} b_{n,p} z^n = B(z) \quad \text{and}$$

$$\frac{\partial B}{\partial y} = \sum_{n,p} p b_{n,p} x^n y^{p-1}$$

Let

$$H(z) = \left.\frac{\partial B}{\partial y}\right|_{(z,1)} = \sum_{n,p} p b_{n,p} z^n = \sum_n h_n z^n, \quad \text{say.}$$

For each n,

$$\frac{h_n}{b_n}$$

is the average pathlength of an n-node binary tree.

Now for a recurrence for $b_{n,p}$ valid for all positive n. The path length is defined as the sum of the node depths.

For example, suppose a binary tree has a left subtree with i nodes and total path length q. The path length is the total depth of nodes within the left subtree; but the depth of a node within the full tree is increased by 1. Therefore these nodes contribute $q+i$ to the total node depth. Hence we get the recurrence, valid for all $n > 0$,

$$b_{n,p} = \sum_{i+j=n-1;i+j+q+r=p} b_{i,q}b_{j,r}$$

Substitute, so to speak, wz for x and w for y, i.e., multiply by $(zw)^i w^q (zw)^j w^r$.

$$\sum_{i+j=n-1;i+j+q+r=p} b_{i,q}(zw)^i w^q b_{j,r}(zw)^j w^r$$
$$= \sum b_{i,q}b_{j,r}z^{n-1}w^p \quad = \quad B^2(zw,w)$$
$$= \frac{B(z,w)-1}{z}$$

So we now have the equation

$$zB^2(zw,w) = B(z,w) - 1$$

Take the partial derivative with respect to w, at $w = 1$. Note that $(x,y) = (zw,w) = (z,1)$ for purposes of differentiation.

Recall
$$H(z) = \left.\frac{\partial B}{\partial y}\right|_{(z,1)} = \sum_{n,p} pb_{n,p}z^n = \sum_n h_n z^n, \quad \text{say.}$$

Differentiate $B(z,w)$ with respect to w at $w = 1$:

$$\frac{\partial B(z,w)}{\partial w} =$$
$$\frac{\partial}{\partial w}\sum_{n,p} b_{n,p}z^n w^p =$$
$$\sum_{n,p} pb_{n,p}z^n = H(z)$$

at $w = 1$, and, again differentiating wrt w at $w = 1$,

$$\frac{\partial}{\partial w}(zB(zw,w)^2) = 2zB(zw,w)\partial B(zw,w);$$
$$\frac{\partial B(zw,w)}{\partial w} =$$
$$\sum_{n,p}(n+p)b_{n,p}z^n w^{n+p-1} =$$
$$\sum_{n,p} nb_{n,p}z^n + \sum_{n,p} pb_{n,p}z^n =$$
$$zB'(z) + H(z)$$

Putting these together, and using the previous formula for $B(z)$, we get, after considerable effort,

$$H(z) = \frac{2z^2 B(z) B'(z)}{1 - 2zB(z)}$$

and

$$H(z) = \frac{2zB(z)}{1 - 2zB(z)}(zB'(z)) = \frac{1 - \sqrt{1-4z}}{\sqrt{1-4z}} \left(\frac{1}{\sqrt{1-4z}} + \frac{\sqrt{1-4z} - 1}{2z} \right).$$

This can be interpreted as

$$\frac{1}{1-4z} + \text{ lower-order terms.}$$

Ignoring the lower-order terms, $h(z) \approx 1/(1-4z)$ and $h_n \approx 4^n$. This gives an estimate of

$$\frac{4^n}{\frac{1}{n+1}\binom{2n}{n}}$$

which, using the Stirling approximation, is roughly

$$(n+1)\sqrt{n\pi}$$

A full analysis would give the average pathlength

$$\frac{4^n}{\frac{1}{n+1}\binom{2n}{n}} - 3n - 1$$

which makes little difference.

(8.1) Corollary *The average path-length in an n-node binary tree is roughly*

$$n\sqrt{n\pi}$$

This is in contrast to the average pathlength $O(n \log n)$ for binary *search* trees formed by inserting keys in random order.