MA 3467 quiz 2 answers: Thursday 17/10/13

A binary tree T is *maximal* if

$$n = 2^{h+1} - 1$$

where n is the number of nodes in T and h is its height.

(1) Estimate in terms of n the average depth of nodes in a maximal tree with n nodes.

Answer. Let h be the height, so $n + 1 = 2^{h+1}$. The general formula

$$\sum_{r=0}^{h} rx^r = \frac{x - (h+1)x^{h+1} + hx^{h+2}}{(1-x)^2}$$

can be found by differentiating a geometric series, or by the method used traditionally in summing a geometric series.

Since $h = \log_2(n+1) - 1$ and $2^{h+1} = n + 1$, substituting 2 for x we get

$$\sum_{r=0}^{h} r2^r = 2 + (h-1)2^{h+1} = 2 + (\log_2(n+1) - 2)(n+1) = (n+1)\log_2(n+1) - 2n.$$

So the **answer** is: the average depth is

$$\frac{n+1}{n}\log_2(n+1) - 2.$$

An approximate solution would also be acceptable. There are $(n+1)/2$ leaves at depth $\log_2(n+1) - 1$, so the average depth is at least

$$\left(1 + \frac{1}{2n}\right)(\log_2(n+1) - 1)$$

and, since every node has height $\leq h$, the average depth is at most

$$\log_2(n+1) - 1$$

This is simplest to express with the Θ-notation. We say that f is $\Theta(g)$ if f is $O(g)$ and g is $O(f)$. The answer can then be expressed as follows: average depth is $\Theta(n \log n)$.

Note. Some people answered the question as follows. Suppose a_h is the average depth

$$\frac{\sum_{r=0}^{h} 2^r r}{2^{h+1} - 1} \approx \frac{a_{h-1}}{2} + \frac{h}{2}$$

from which it was concluded that a_h is $O(h)$. I don't see that this is obvious. There is a standard approach to these recurrences (linear constant coefficient): variation of parameters, which leads one to substitute $a_h = b_h/2^h$:

$$b_h - b_{h-1} = h2^h.$$

This means $b_h = C + \sum r2^r$, and we need to sum the same series as before.

On the other hand, one could unwind the recurrence

$$a_h = a_{h-1}/2 + h/2 = a_{h-2}/4 + (h-1)/4 + h/2 \ldots$$

arriving at the result $a_h \leq h$. This may be closer to making the conclusion obvious.

(2) Estimate the average height of nodes in a maximal tree with n nodes.

Answer.

$$\sum_{r=0}^{h} r2^{h-r} = 2^h \frac{\frac{1}{2} - (h+1)\left(\frac{1}{2}\right)^{h+1} + h2^{h+2}}{(1/2)^2} = 2^{h+1} - h - 2 = n - h - 1$$

so the average is $1 - (h+1)/n$ or $\Theta(1)$.

An easier estimate uses

$$\sum_{0}^{\infty} rx^r = x\frac{d}{dx}\frac{1}{1-x} = \frac{x}{(1-x)^2}$$

or, 2 when $x = 1/2$. This yields the estimate

$$2^h(2)$$

so the average is $< 1/(1-(1/2)^{h+1}) \approx 1/(1-1/n)$.

Again, some students approached it differently, but I forget exactly how.

(3) In a binary tree T, define the Q-value (ad-hoc name) at any node as

$$\sum_{y \text{ ancestor of } x} |\text{descendants of } y|$$

and let the Q-value of a tree T be the total Q value of its nodes. Estimate the *average* Q-value of binary search trees T. The general idea is as in §4.7.

Answer. Say a binary tree T has $n+1$ nodes and left and right subtrees T_1 and T_2. The if a node in T_1 has Q-value q in T_1, then that increases by $n+1$ as Q-value in T, since the root has $n+1$ descendants. Also the root has Q-value $n+1$. Therefore

$$Q(T) = Q(T_1) + Q(T_2) + (n+1)^2.$$

Following the methods of $\S 4.7$, we arrive at the following recurrence for the average Q-value, call it Q

$$Q(n+1) = (n+1)^2 + \frac{2}{n+1} \sum_{i=0}^{n} A(i)$$

To make it simpler, I prefer to consider the following over-estimate

$$\begin{aligned} Q(n+1) &= (n+3)(n+2) + \dots \\ (n+1)Q(n+1) &= (n+3)(n+2)(n+1) + 2\sum_{i=0}^{n} Q(i) \\ nQ(n) &= (n+2)(n+1)(n) + 2\sum_{i=0}^{n-1} Q(i) \\ (n+1)Q(n+1) - nQ(n) &= 3(n+2)(n+1) + 2Q(n) \\ (n+1)Q(n+1) - (n+2)Q(n) &= 3(n+2)(n+1) \\ \frac{Q(n+1)}{n+2} - \frac{Q(n)}{n+1} &= 3 \\ Q(n)/(n+1) &= C + 3n \\ Q(n) &= (C+3n)(n+1) \end{aligned}$$

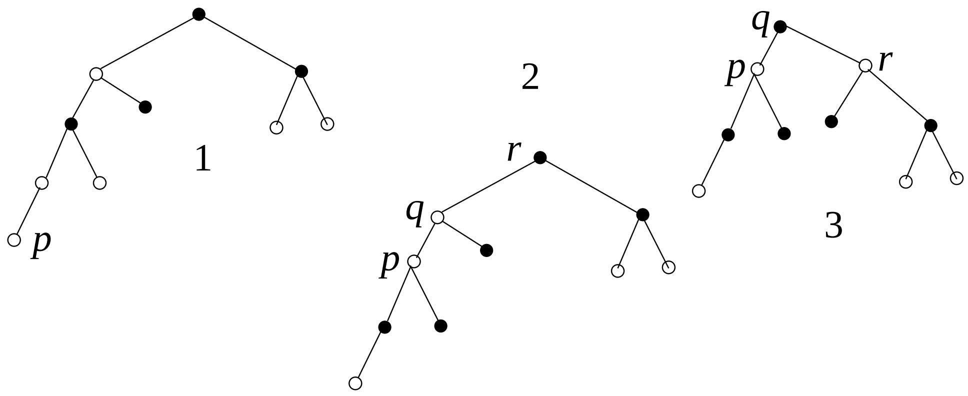
How big is the over-estimate? We can write a recurrence for the difference, call it D.

$$\begin{aligned} (n+3)(n+2) - (n+1)^2 &= 3n+5 \\ (n+1)D(n+1) - nD(n) &= (n+1)(3n+5) - n(3n+2) - 2D(n) \\ (n+1)D(n+1) - (n+2)D(n) &= 6n+5 \end{aligned}$$

whence $D(n)$ is $O(n \log n)$: $Q(n) = (C + 3n - O(\log n))(n+1)$.

(4) Insert a new key leftmost in the red-black tree below. Follow the web notes *scrupulously* and be careful.

Answer.



(5) Delete the leftmost node in the red-black tree below. Again, be scrupulous and be careful.

Answer.