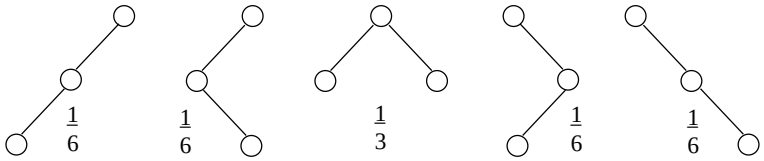
The Expected Height of Random Search Trees.

A *random search tree* for a set S of n keys (say, natural numbers, pairwise distinct) is recursively built as follows: If S is empty, then the tree is empty (i.e., has no vertex at all). Otherwise, we choose a random key w from S (each key in S with probability $\frac{1}{n}$), we create a root vertex which stores w, and two pointers to a left child (which is a random search tree for $S_{<w} := \{k \in S \mid k < w\}$), and a right child (which is a random search tree for $S_{>w} := \{k \in S \mid k > w\}$); if $S_{<w}$ or $S_{>w}$ is empty, then the respective child does not exist. The following figure displays all search trees for three keys, each with its probability to occur as a random search tree.



A number of expected quantities are quite easy to derive for such a random search tree (if you know how, [1]). For example the expected depth of the smallest key in the tree is $H_n - 1$, where $H_n := 1 + \frac{1}{2} + \cdots + \frac{1}{n}$ is the n-th harmonic number; in general, the expected depth of the k-smallest key (we call this the key of *rank* k) is $H_{n-k+1} + H_k - 2$. The expected sum of the depth of all n keys is $2(n+1)H_n - 4n$.

What we are interested in here is the expected maximum depth of such a tree, i.e. the height of a random search tree for n keys. The following describes a proof of an upper bound for this quantity – surprisingly enough (after having seen the proof), this bound is already tight as for the constant in the leading term.

So, for $1 \le i \le n$, let $Y_n^{(i)}$ be the random variable for the depth of the key of rank i in a random search tree for n keys. Then $X_n := \max\{Y_n^{(1)}, Y_n^{(2)}, \ldots, Y_n^{(n)}\}$ is the random variable for the height of the tree with n keys.

We can now write

$$\mathsf{E}(X_n) \le \log \mathsf{E}(2^{X_n}) = \log \mathsf{E}(2^{\max_{1 \le i \le n} Y_n^{(i)}}) < \log \mathsf{E}\Big(\sum_{1 \le i \le n,\ i \text{ is leaf}} 2^{Y_n^{(i)}}\Big) , \tag{1}$$

with log the logarithm of base 2. The first inequality uses Jensen's inequality which states that $f(\mathsf{E}(X)) \le \mathsf{E}(f(X))$ for any convex function f (provided the expectations exist). The gain is that we can now estimate a random variable involving 'max' by the random variable $Z_n := \sum_{1 \le i \le n,\ i \text{ is leaf}} 2^{Y_n^{(i)}}$ involving a sum. The conditional part 'i is leaf' causes no problems as we shall see; in fact, we put it there to make life easier.

Obviously, $\mathsf{E}(Z_0) = 0$ and $\mathsf{E}(Z_1) = 1$. There are two trees on two vertices, each one appears with probability $\frac{1}{2}$, which gives $\mathsf{E}(Z_2) = 2$; by checking the figure above for the case of three keys, we obtain $\mathsf{E}(Z_3) = 4$. Enough of fiddling around with small values. For $n \ge 2$, we get

$$\mathsf{E}(Z_n) = \frac{1}{n} \sum_{1 \le i \le n} \mathsf{E}(Z_n \mid \text{ root has rank } i) ,$$

where $\mathsf{E}(Z_n \mid \text{ root has rank } i) = 2(\mathsf{E}(Z_{i-1}) + \mathsf{E}(Z_{n-i}))$. Putting $z_n := \mathsf{E}(Z_n)$, we have $z_0 = 0$, $z_1 = 1$ and, for $n \ge 2$,

$$z_n = \frac{4}{n} \sum_{0 \le j \le n-1} z_j . \tag{2}$$

Consequently, for $n \geq 3$, $nz_n - (n-1)z_{n-1} = 4z_{n-1}$, and so $nz_n = (n+3)z_{n-1}$ or

$$\frac{z_n}{(n+3)(n+2)(n+1)} = \frac{z_{n-1}}{(n+2)(n+1)n} = \cdots = \frac{z_2}{5 \cdot 4 \cdot 3} = \frac{1}{30} \ .$$

That is, $z_n = \frac{(n+3)(n+2)(n+1)}{30}$ for $n \geq 2$.

If we plug that bound into (1), then we get an upper bound of $3 \log n + O(1) \approx 4.32808 \ln n + O(1)$ for the expected height of a random search tree. Does this give already the right constant? Well, contrary to what we announced before, this is not the case – *yet*! But we still have the base 2 to play with.

We set $Z_n := \sum_{1 \leq i \leq n,\ i \text{ is leaf}} C^{Y_n^{(i)}}$, with C some real number greater than 1. Similar to (1), we have $\mathsf{E}(X_n) \leq \log_C \mathsf{E}(Z_n)$. The recursion for $z_n := \mathsf{E}(Z_n)$ is the same as the one given in (2), except that 4 gets replaced by $2C$, which eventually leads to $nz_n = (n + 2C - 1)z_{n-1}$ for $n \geq 2$. This yields

$$z_n = (1 + \frac{2C-1}{n})z_{n-1} = (1 + \frac{2C-1}{n})(1 + \frac{2C-1}{n-1}) \cdots (1 + \frac{2C-1}{3})z_2 \ .$$

Since $z_2 = C \leq (1 + \frac{2C-1}{2})$, and $1 + x \leq e^x$ for any real number x, this implies

$$z_n \leq e^{(2C-1)\sum_{i=2}^{n}(1/i)} = e^{(2C-1)(H_n - 1)} < e^{(2C-1)\ln n} = n^{2C-1} \ . \tag{3}$$

Invoking (1) (in its adopted version with 'C' instead of '2'), this gives a bound of

$$\mathsf{E}(X_n) < \frac{2C-1}{\ln C} \ln n \ .$$

This bound attains its extremal values if $2 \ln C - 2 + 1/C = 0$, or, equivalently, if $(\frac{e}{C})^{2C} = e$. Note that for these values of C, we have $\frac{2C-1}{\ln C} = 2C$. Hence, we have shown

Theorem 1 *The expected height of a random search tree for n keys is bounded by $c \ln n$, where $c \approx 4.311070\ldots$ is the unique value greater than 2 which satisfies $(\frac{2e}{c})^c = e$.* □

The constant in the leading term is already tight, as it was shown by Devroye [2]; the proof of this fact is considerably more involved than the upper bound proof we have just seen. The upper bound has been shown before by Robson.

Let us conclude by pointing out that knowing a good estimate for $\mathsf{E}(C^{X_n})$ immediately gives a good tail estimate for X_n via Markov's inequality, namely

$$\mathsf{Pr}(X_n > \tau \ln n) = \mathsf{Pr}(C^{X_n} > C^{\tau \ln n}) = \mathsf{Pr}(C^{X_n} > \frac{C^{\tau \ln n}}{\mathsf{E}(C^{X_n})}\mathsf{E}(C^{X_n})) \leq$$

$$\frac{\mathsf{E}(C^{X_n})}{C^{\tau \ln n}} < n^{2C-1-\tau \ln C} \ .$$

$2C - 1 - \tau \ln C$ is minimized for $C = \tau/2$.

Theorem 2

$$\mathsf{Pr}(X_n > \tau \ln n) < n^{\tau(1-\ln(\tau/2))-1} \ ;$$

in particular $\mathsf{Pr}(X_n > 2e \ln n) < \frac{1}{n}$. □

If we set $\tau = c$, where c is the constant in Theorem 1, then we obtain $\mathsf{Pr}(X_n > c \ln n) < 1$, which we might have guessed before. However, you may want to reconsider the estimate obtained in (3) to get a nontrivial (though still constant) bound for this probability.

[1] Aragon, C., and Seidel, R., Randomized search trees, FOCS '91.

[2] Devroye, L. A note on the height of binary search trees, *J. ACM* **33** (1986) 489–498.

[3] Vitter, J.S., and Flajolet, Ph., Average-case analysis of algorithms and data structures, *in* "*Handbook of Theoretical Computer Science*", Volume A (1990) 431–520.