THE m-VERSION OF BINARY SEARCH TREES: AN AVERAGE-CASE ANALYSIS

HELMUT PRODINGER

Abstract. Following a suggestion of Cichoń and Macyna, binary search trees are generalized by keeping m (classical) binary search trees and distributing incoming data at random to the individual trees. Costs for unsuccessful resp. successful search are analyzed, as well as the internal path length.

1. Introduction

Cichoń, together with his coauthor Macyna, had the seminal idea [2] to generalize *approximate counting* to *approximate counting with m counters.* While in the original version [3] a stream of letters (a word) is dealt with a counter in a certain way (that is of no interest here), the new version uses m counters, and chooses for each letter one of these counters (with probability $\frac{1}{m}$) where it is dealt with as usual. The *result* of the procedure is the sum of the individual results of the m counters.

This fundamental idea should, however, not be restricted to approximate counting! Indeed, it can be considered within a variety of different contexts. In this paper, the fundamental idea is applied to *binary search trees.* They are very well understood and described in classic books such as [7] and [9], with plenty of backward pointers to the older literature due to Lynch, Hibbard, Louchard, Brown, Shubert, and many others. We assume that, instead of just one, m binary search trees are kept, and for each element, when inserting it, a decision is made, to which of the m trees it is being sent. Of course, for algorithmic purposes, this choice must be deterministic, so that one knows, in which tree to search. However, for the analysis, it is assumed that each tree is equally likely, and will be selected with probability $\frac{1}{m}$.

Almost all the information about binary search trees that is known to this day can be found in the encyclopedic books [7, 9]. We only mention that they originate from random permutations (typically of $\{1, \dots, n\}$); a new element is compared to the root, and, if there is no space for it, moved to the left/right if it is smaller/larger than the root; then the process continues. A binary search tree is used as a data structure. It is thus essential that one can find existing elements in a reasonable number of steps, and also get the information that a searched element is not present after a small number of comparisons.

This is the first paper about m-binary search trees, and the hope is that many more will be written in the future, by various specialists. Thus, no completeness is aimed at. Three parameters are studied: The cost for inserting a new element into the m trees, which is related to the cost of unsuccessful search; then the cost for successful search, which is the average of the level of all the elements in all m trees, and then the internal path length, which is the sum of the internal path lengths of the m trees.

Date: August 27, 2012.

In the classical case, probability generating functions are available, so that one can extract moments from them, which can be written in terms of harmonic numbers and generalizations. We describe here how these probability generating functions translate to the m-model.

We try to use consistent notation: If the probability generating function is $f(z)$, then we write $\mathcal{F}(z)$ for the transformed m-version. We always write n for the number of nodes in a classical binary search tree and N for the total number of nodes in the m binary search trees. Furthermore, we write $P_{n,k}$, $\mathcal{P}_{N,k}$, E_n, $E_n^{(2)}$, $\mathcal{E}_N$, $\mathcal{E}_N^{(2)}$, for probabilities and moments. We use the second factorial moments on our way to the variance.

A crucial expression is

$$m^{-N}\frac{N!}{n_1!\dots n_m!},$$

with $n_1+\dots+n_m$, which is the probability that the N data split into m sets of sizes $n_1,\dots,n_m$ each.

It turns out that we have to use three auxiliary quantities, named S_N, T_N, U_N, which are introduced in the next section. All our quantities of interest can be expressed in terms of them. This is done in full in the section on unsuccessful search, but only sketched in the remaining sections, since the actual computations are quite long.

The intuition is of course that each of the m binary search tree should have roughly N/m nodes; the analysis that follows will make this precise.

The classic book [8] is an excellent source on harmonic numbers and their manipulation; in fact, quantity T_n appears already in it!

2. Unsuccessful search

The first parameter that we study is the number of comparisons to insert node n into a binary search tree with n nodes. This is directly related to searching for a key which is not present, since it is equivalent to insert this (nonexistent) item as the $(n+1)$-st node. The probability generating function is

$$g_n(z)=\frac{1}{n!}2z(2z+1)\dots(2z+n-2),$$

so that the probability that k comparisons are needed, is

$$P_{n,k}=\frac{1}{n!}[z^k]2z(2z+1)\dots(2z+n-2).$$

From this, one derives

$$E_n=g_n'(1)=\sum_{j=0}^{n-2}\frac{2}{j+2}=2\sum_{j=2}^{n}\frac{2}{j}=2(H_n-1)$$

and

$$E_n^{(2)}=8\sum_{2\le i<j\le n}\frac{1}{ij}=8\sum_{1\le i<j\le n}\frac{1}{ij}-8\sum_{1<j\le n}\frac{1}{j}=4H_n^2-4H_n^{(2)}-8H_n+8.$$

All this is classical. Now we translate this into the m-model. The largest node N sits in one of the m binary search trees of size n. Therefore

$$\begin{aligned}
\mathcal{P}_{n,k} &= m^{-N} \sum_{n_1+\dots+n_m=N} \frac{N!}{n_1!\dots n_m!} \sum_{i=1}^{m} \frac{\binom{N-1}{n_i-1}}{\binom{N}{n_i}} P_{n_i,k} \\
&= m^{1-N} \sum_{n_1+\dots+n_m=N} \frac{N!}{n_1!\dots n_m!} \frac{n_1}{N} P_{n_1,k} \\
&= m^{1-N} \sum_{n=1}^{N} \binom{N}{n} (m-1)^{N-n} \frac{n}{N} P_{n,k} \\
&= m^{1-N} \sum_{n=1}^{N} \binom{N-1}{n-1} (m-1)^{N-n} P_{n,k}.
\end{aligned}$$

The quotient $\binom{N-1}{n_i-1}/\binom{N}{n_i}$ is the probability that the remaining n_i-1 nodes can be chosen. On the level of probability generating functions, this means

$$\mathcal{G}_n(z) = m^{1-N} \sum_{n=1}^{N} \binom{N-1}{n-1} (m-1)^{N-n} g_n(z). \tag{2.1}$$

The last form is obtained by multiplication by z^k and summing. Moments can be computed from this, using differentiations. In order to do so, we need some auxiliary sums, that will be also useful in later sections.

Lemma 1. (1)

$$\begin{aligned}
S_n &= \sum_{k=1}^{n} \binom{n}{k} x^k \frac{1}{k} \\
&= \sum_{k=1}^{n} \frac{(1+x)^k}{k} - H_n.
\end{aligned}$$

(2)

$$\begin{aligned}
T_n &= \sum_{k=1}^{n} \binom{n}{k} x^k H_k \\
&= H_n(1+x)^n - (1+x)^n \sum_{k=1}^{n} \frac{1}{k(1+x)^k}.
\end{aligned}$$

(3)

$$\begin{aligned}
U_n &= \sum_{k=1}^{n} \binom{n}{k} x^k \sum_{1\le i<j\le k} \frac{1}{ij} \\
&= (1+x)^n \frac{H_n^2 - H_n^{(2)}}{2} - (1+x)^n \sum_{k=1}^{n} \frac{1}{k(1+x)^k} H_{n-k} \\
&+ (1+x)^n \sum_{k=1}^{n} \frac{1}{k(1+x)^k} H_k - (1+x)^n \sum_{k=1}^{n} \frac{1}{k^2(1+x)^k}.
\end{aligned}$$

Proof. All the proofs are using the basic recursion for binomial coefficients, to create a first order recursion which can be solved by summation. The procedure for T_n is contained in [8].

$$\begin{aligned}
S_{n+1} &= \sum_{k=1}^{n+1} \left[\binom{n}{k} + \binom{n}{k-1}\right] x^k \frac{1}{k} \\
&= S_n + \sum_{k=1}^{n+1} \binom{n}{k-1} x^k \frac{1}{k} \\
&= S_n + \frac{1}{n+1} \sum_{k=1}^{n+1} \binom{n+1}{k} x^k \\
&= S_n + \frac{1}{n+1}(1+x)^{n+1} - \frac{1}{n+1}.
\end{aligned}$$

$$\begin{aligned}
T_{n+1} &= \sum_{k=1}^{n+1} \left[\binom{n}{k} + \binom{n}{k-1}\right] x^k H_k \\
&= T_n + \sum_{k=1}^{n+1} \binom{n}{k-1} x^k H_{k-1} + \sum_{k=1}^{n+1} \binom{n}{k-1} x^k \frac{1}{k} \\
&= T_n + x \sum_{k=0}^{n} \binom{n}{k} x^k H_k + \frac{1}{n+1} \sum_{k=1}^{n+1} \binom{n+1}{k} x^k \\
&= (1+x)T_n + \frac{1}{n+1}(1+x)^{n+1} - \frac{1}{n+1}.
\end{aligned}$$

$$\begin{aligned}
U_{n+1} &= U_n + \sum_{k=1}^{n+1} \binom{n}{k-1} x^k \sum_{1\le i<j\le k} \frac{1}{ij} \\
&= U_n + x \sum_{k=0}^{n} \binom{n}{k} x^k \sum_{1\le i<j\le k+1} \frac{1}{ij} \\
&= U_n + x \sum_{k=0}^{n} \binom{n}{k} x^k \sum_{1\le i<j\le k} \frac{1}{ij} + x \sum_{k=0}^{n} \binom{n}{k} x^k \sum_{1\le i\le k} \frac{1}{i(k+1)} \\
&= (1+x)U_n + \frac{1}{n+1} \sum_{k=0}^{n} \binom{n+1}{k+1} x^{k+1} H_k \\
&= (1+x)U_n + \frac{1}{n+1} \sum_{k=1}^{n+1} \binom{n+1}{k} x^k (H_k - \frac{1}{k}) \\
&= (1+x)U_n + \frac{1}{n+1} \sum_{k=1}^{n+1} \binom{n+1}{k} x^k H_k - \frac{1}{n+1} \sum_{k=1}^{n+1} \binom{n+1}{k} x^k \frac{1}{k}
\end{aligned}$$

$$= (1+x)U_n + \frac{1}{n+1}T_{n+1} - \frac{1}{n+1}S_{n+1}.$$

Therefore

$$\begin{aligned}
\frac{U_n}{(1+x)^n} &= \frac{U_{n-1}}{(1+x)^{n-1}} + \frac{1}{n(1+x)^n}T_n - \frac{1}{n(1+x)^n}S_n \\
&= \frac{U_{n-1}}{(1+x)^{n-1}} + \frac{1}{n}H_n - \frac{1}{n}\sum_{k=1}^{n}\frac{1}{k(1+x)^k} - \frac{1}{n(1+x)^n}\sum_{k=1}^{n}\frac{(1+x)^k}{k} + \frac{1}{n(1+x)^n}H_n \\
&= \sum_{j=1}^{n}\frac{1}{j}H_j - \sum_{j=1}^{n}\frac{1}{j}\sum_{k=1}^{j}\frac{1}{k(1+x)^k} - \sum_{j=1}^{n}\frac{1}{j(1+x)^j}\sum_{k=1}^{j}\frac{(1+x)^k}{k} + \sum_{j=1}^{n}\frac{1}{j(1+x)^j}H_j \\
&= \frac{H_n^2+H_n^{(2)}}{2} - \sum_{k=1}^{n}\sum_{j=k}^{n}\frac{1}{k(1+x)^k}\frac{1}{j} - \sum_{k=1}^{n}\sum_{j=k}^{n}\frac{(1+x)^k}{k}\frac{1}{j(1+x)^j} + \sum_{j=1}^{n}\frac{1}{j(1+x)^j}H_j \\
&= \frac{H_n^2+H_n^{(2)}}{2} - \sum_{k=1}^{n}\frac{1}{k(1+x)^k}(H_n - H_{k-1}) \\
&\quad - \sum_{k=1}^{n}\sum_{j=0}^{n-k}\frac{1}{(j+k)k(1+x)^j} + \sum_{j=1}^{n}\frac{1}{j(1+x)^j}H_j \\
&= \frac{H_n^2+H_n^{(2)}}{2} - H_n\sum_{k=1}^{n}\frac{1}{k(1+x)^k} + \sum_{k=1}^{n}\frac{1}{k(1+x)^k}H_k - \sum_{k=1}^{n}\frac{1}{k^2(1+x)^k} \\
&\quad - \sum_{k=1}^{n}\sum_{j=1}^{n-k}\frac{1}{(j+k)k(1+x)^j} - \sum_{k=1}^{n}\frac{1}{k^2} + \sum_{j=1}^{n}\frac{1}{j(1+x)^j}H_j \\
&= \frac{H_n^2-H_n^{(2)}}{2} - H_n\sum_{k=1}^{n}\frac{1}{k(1+x)^k} + 2\sum_{k=1}^{n}\frac{1}{k(1+x)^k}H_k - \sum_{k=1}^{n}\frac{1}{k^2(1+x)^k} \\
&\quad - \sum_{j=1}^{n-1}\sum_{k=1}^{n-j}\frac{1}{jk(1+x)^j} + \sum_{j=1}^{n-1}\sum_{k=1}^{n-j}\frac{1}{(j+k)j(1+x)^j} \\
&= \frac{H_n^2-H_n^{(2)}}{2} - H_n\sum_{k=1}^{n}\frac{1}{k(1+x)^k} + 2\sum_{k=1}^{n}\frac{1}{k(1+x)^k}H_k - \sum_{k=1}^{n}\frac{1}{k^2(1+x)^k} \\
&\quad - \sum_{j=1}^{n-1}\frac{1}{j(1+x)^j}H_{n-j} + \sum_{j=1}^{n-1}\frac{1}{j(1+x)^j}(H_n - H_j) \\
&= \frac{H_n^2-H_n^{(2)}}{2} + \sum_{k=1}^{n}\frac{1}{k(1+x)^k}H_k - \sum_{k=1}^{n}\frac{1}{k^2(1+x)^k} - \sum_{k=1}^{n}\frac{1}{k(1+x)^k}H_{n-k}.
\end{aligned}$$

□

In our applications, $x = \frac{1}{m-1}$, and then the formulæ read:

$$\Big(1-\frac{1}{m}\Big)^N T_N = H_N - \sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k,$$

$$\begin{aligned}\Big(1-\frac{1}{m}\Big)^N U_N &= \frac{H_N^2-H_N^{(2)}}{2}\\ &\quad -\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_{N-k}+\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_k-\sum_{k=1}^{N}\frac{1}{k^2}\Big(1-\frac{1}{m}\Big)^k.\end{aligned}$$

After these long but necessary computations have been done, we can now compute the moments:

$$\begin{aligned}\mathcal{E}_N &= m^{1-N}\sum_{n=1}^{N}\binom{N-1}{n-1}(m-1)^{N-n}2(H_n-1)\\ &= 2m^{1-N}\sum_{n=1}^{N}\binom{N-1}{n-1}(m-1)^{N-n}H_n-2\\ &= 2m^{1-N}\sum_{n=1}^{N}\binom{N-1}{n-1}(m-1)^{N-n}(H_{n-1}+\frac{1}{n})-2\\ &= 2m^{1-N}(m-1)^{N-1}T_{N-1}+\frac{2m}{N}-\frac{2m}{N}\Big(1-\frac{1}{m}\Big)^N-2.\end{aligned}$$

Therefore

$$\mathcal{E}_N = 2H_{N-1}-2\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k+\frac{2m}{N}-\frac{2m}{N}\Big(1-\frac{1}{m}\Big)^N-2.$$

Now, by two differentiations, we find by a similar (but much longer) computation as before

$$\begin{aligned}\mathcal{E}_N^{(2)}+4\mathcal{E}_N &= m^{1-N}\sum_{n=1}^{N}\binom{N-1}{n-1}(m-1)^{N-n}[4H_n^2-4H_n^{(2)}]\\ &= 8\Big(1-\frac{1}{m}\Big)^{N-1}U_{N-1}+\frac{8m}{N}\Big(1-\frac{1}{m}\Big)^N T_N-\frac{8m}{N}\Big(1-\frac{1}{m}\Big)^N S_N\\ &= 4(H_{N-1}^2-H_{N-1}^{(2)})\\ &\quad -8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_{N-1-k}+8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_k-8\sum_{k=1}^{N-1}\frac{1}{k^2}\Big(1-\frac{1}{m}\Big)^k\\ &\quad +\frac{8m}{N}H_N-\frac{8m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k\\ &\quad -\frac{8m}{N}\sum_{k=0}^{N-1}\frac{1}{N-k}\Big(1-\frac{1}{m}\Big)^k+\frac{8m}{N}\Big(1-\frac{1}{m}\Big)^N H_N.\end{aligned}$$

From these results, we can get the variance *explicitly* as $(\mathcal{E}_N^{(2)}+4\mathcal{E}_N)-3\mathcal{E}_N-(\mathcal{E}_N)^2$. We don't display it since it is quite long. However, we will drop exponentially small terms of the form $O(\varrho^N)$ with $1-\frac{1}{m}<\varrho<1$; then the results are a bit more appealing:

$$\mathcal{E}_N \sim 2H_{N-1}-2\log m+\frac{2m}{N}-2;$$

$$\mathcal{E}_N^{(2)} + 4\mathcal{E}_N \sim 4(H_{N-1}^2 - H_{N-1}^{(2)}) - 8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_{N-1-k} + 8C_1(m) - 8C_2(m)$$
$$+\frac{8m}{N}H_N - \frac{8m}{N}\log m - \frac{8m}{N}\sum_{k=0}^{N-1}\frac{1}{N-k}\Big(1-\frac{1}{m}\Big)^k.$$

with

$$C_1(m) = \sum_{k\ge1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_k \quad\text{and}\quad C_2(m) = \sum_{k\ge1}\frac{1}{k^2}\Big(1-\frac{1}{m}\Big)^k.$$

(The sums can be extended to infinity; the extra terms are absorbed in our exponentially small remainder term.)

The remaining sums can be asymptotically evaluated:

$$\begin{aligned}\sum_{k=0}^{N-1}\frac{1}{N-k}\Big(1-\frac{1}{m}\Big)^k &= [z^N]\log\frac{1}{1-z}\cdot\frac{1}{1-z(1-\frac{1}{m})}\\ &= [z^N]\log\frac{1}{1-z}\cdot\Big[m - m(m-1)(1-z)+\cdots\Big]\\ &= \frac{m}{N} - m(m-1)\Big(\frac{1}{N}-\frac{1}{N-1}\Big) + O(N^{-3});\end{aligned}$$

$$\begin{aligned}&\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k H_{N-1-k} = [z^{N-1}]\sum_{k\ge1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k z^k\cdot\sum_{k\ge1}H_k z^k\\ &= [z^{N-1}]\log\frac{1}{1-z(1-\frac{1}{m})}\frac{1}{1-z}\log\frac{1}{1-z}\\ &= [z^{N-1}]\Big[\log m - (m-1)(1-z) + \frac{1}{2}(m-1)^2(1-z)^2+\cdots\Big]\frac{1}{1-z}\log\frac{1}{1-z}\\ &= (\log m)H_{N-1} - (m-1)\frac{1}{N-1} + \frac{1}{2}(m-1)^2\Big(\frac{1}{N-1}-\frac{1}{N-2}\Big) + O(N^{-3}).\end{aligned}$$

Not more is required than the generating function of the harmonic numbers. What we have done here is justified by *singularity analysis*, as described in [4]; note that $z=1$ is the dominant singularity here.

Theorem 1. *The expectation and variance of the number of comparisons needed to insert the last element into m binary search trees of altogether N nodes, are given by*

$$\mathcal{E}_N = 2\log\frac{N}{m} + 2\gamma - 2 + O\Big(\frac{1}{N}\Big),$$
$$\mathcal{V}_N = 2\log\frac{N}{m} - 4\log^2 m + 2\gamma + 2 - \frac{2}{3}\pi^2 + 8C_1(m) - 8C_2(m) + O\Big(\frac{1}{N}\Big).$$

More terms in the asymptotic expansions are easily available.

3. Successful search

Now we look at successful search in binary search trees. The model is that the comparisons to find all possible nodes are *added*, and this count is than divided by the total number of nodes. This parameter has the following probability generating function:

$$R_n(z)=\frac{z(2z)(2z+1)\dots(2z+n-1)}{nn!(2z-1)}-\frac{z}{n(2z-1)}.$$

It translates into the m-model as follows:

$$\begin{aligned}\mathcal{R}_N(z)&=m^{-N}\sum_{n_1+\dots+n_m=N}\binom{N}{n_1,\dots,n_m}\frac{n_1R_{n_1}(z)+\dots+n_mR_{n_m}(z)}{N}\\&=m^{-N}\frac{1}{N}\sum_{n=1}^{N}\binom{N}{n}(m-1)^{N-n}nR_n(z);\end{aligned}$$

note that we add the comparisons in each subtree, given by $n_iR_{n_i}(z)$, and then divide by the total number N. The following results are classical:

$$\begin{aligned}nR_n'(1)=nE_n&=2(n+1)H_n-3n,\\nR_n''(1)=nE_n^{(2)}&=4(n+1)(H_n^2-H_n^{(2)})-12nH_n-4H_n+16n.\end{aligned}$$

Consequently we can evaluate moments, in the same style as in the last section. We don't present all the long computations here.

$$\begin{aligned}\mathcal{E}_N&=m^{1-N}\frac{1}{N}\sum_{n=1}^{N}\binom{N}{n}(m-1)^{N-n}[2(n+1)H_n-3n]\\&=2\Big(1-\frac{1}{m}\Big)^{N-1}T_{N-1}+\frac{2m}{N}-\frac{2m}{N}\Big(1-\frac{1}{m}\Big)^{N}+\frac{2m}{N}\Big(1-\frac{1}{m}\Big)^{N}T_N-3\\&=2H_{N-1}-2\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^{k}+\frac{2m}{N}-\frac{2m}{N}\Big(1-\frac{1}{m}\Big)^{N}-3\\&+\frac{2m}{N}H_N-\frac{2m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^{k}.\end{aligned}$$

Further,

$$\begin{aligned}\mathcal{E}_N^{(2)}&=m^{1-N}\frac{1}{N}\sum_{n=1}^{N}\binom{N}{n}(m-1)^{N-n}nR_n''(1)\\&=m^{1-N}\frac{1}{N}\sum_{n=1}^{N}\binom{N}{n}(m-1)^{N-n}[4(n+1)(H_n^2-H_n^{(2)})-12nH_n-4H_n+16n]\\&=8\Big(1-\frac{1}{m}\Big)^{N-1}U_{N-1}+\frac{4m}{N}\Big(1-\frac{1}{m}\Big)^{N}T_N-\frac{8m}{N}\Big(1-\frac{1}{m}\Big)^{N}S_N\\&+\frac{8m}{N}\Big(1-\frac{1}{m}\Big)^{N}NU_N-12\Big(1-\frac{1}{m}\Big)^{N-1}T_{N-1}-\frac{12m}{N}+\frac{12m}{N}\Big(1-\frac{1}{m}\Big)^{N}+16\\&=4(H_{N-1}^2-H_{N-1}^{(2)})+\frac{4m}{N}(H_N^2-H_N^{(2)})\\&-8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^{k}H_{N-1-k}+8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^{k}H_k-8\sum_{k=1}^{N-1}\frac{1}{k^2}\Big(1-\frac{1}{m}\Big)^{k}\end{aligned}$$

$$
\begin{aligned}
&+\frac{4m}{N}H_N-\frac{4m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k\\
&-\frac{8m}{N}\sum_{k=1}^{N-1}\frac{1}{N-k}\Big(1-\frac{1}{m}\Big)^k-\frac{8m}{N^2}+\frac{8m}{N}H_N\Big(1-\frac{1}{m}\Big)^N\\
&-\frac{8m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^kH_{N-k}+\frac{8m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^kH_k-\frac{8m}{N}\sum_{k=1}^{N}\frac{1}{k^2}\Big(1-\frac{1}{m}\Big)^k\\
&-12H_{N-1}+12\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k-\frac{12m}{N}+\frac{12m}{N}\Big(1-\frac{1}{m}\Big)^N+16.
\end{aligned}
$$

Once again, we drop exponentially small terms to get shorter formulæ:

$$
\mathcal{E}_N\sim 2H_{N-1}-2\log m+\frac{2m}{N}-3+\frac{2m}{N}H_N-\frac{2m}{N}\log m,
$$

$$
\begin{aligned}
\mathcal{E}_N^{(2)}\sim{}&4(H_{N-1}^2-H_{N-1}^{(2)})+\frac{4m}{N}(H_N^2-H_N^{(2)})\\
&-8\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^kH_{N-1-k}+8C_1(m)-8C_2(m)\\
&+\frac{4m}{N}H_N-\frac{4m}{N}\log m-\frac{8m}{N}\sum_{k=0}^{N-1}\frac{1}{N-k}\Big(1-\frac{1}{m}\Big)^k\\
&-\frac{8m}{N}\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^kH_{N-k}+\frac{8m}{N}C_1(m)-\frac{8m}{N}C_2(m)\\
&-12H_{N-1}+12\log m-\frac{12m}{N}+16.
\end{aligned}
$$

The asymptotic form is now computed as in the previous section.

Theorem 2. *The expectation and variance of the number of comparisons in a successful search related to m binary search trees of altogether N nodes, are given by*

$$
\mathcal{E}_N=2\log\frac{N}{m}-3+2\gamma+O\Big(\frac{1}{N}\Big),
$$

$$
\mathcal{V}_N=2\log\frac{N}{m}-4\log^2 m+4-\frac{2}{3}\pi^2+2\gamma+8C_1(m)-8C_2(m)+O\Big(\frac{1}{N}\Big).
$$

More terms in the asymptotic expansions are easily available.

4. Internal path length

The last parameter that we study is the (internal) path length, namely the sum of the distances of all the nodes to the root (in the classical case). In the m-version, it is simply the sum of the path lengths in the m individual trees. It is known that the probability generating functions satisfy

$$
g_n(z)=\frac{z^{n-1}}{n}\sum_{k=1}^{n}g_{k-1}(z)g_{n-k}(z),\quad g_0(z)=1,
$$

whence

$$\begin{aligned}\mathcal{G}_N(z)&=m^{-N}\sum_{n_1+\dots+n_m=N}\binom{N}{n_1,\dots,n_m}\big(g_{n_1}(z)+\dots+g_{n_m}(z)\big)\\&=m^{-N}\sum_{n=0}^{N}\binom{N}{n}(m-1)^{N-n}g_n(z).\end{aligned}$$

It is known that

$$\begin{aligned}g_n'(1)&=2(n+1)H_n-4n,\\g_n''(1)&=4(n+1)^2(H_n^2-H_n^{(2)})-4(n+1)(4n+1)H_n+n(23n+17).\end{aligned}$$

Therefore (and again, the extremely long computations are not displayed)

$$\begin{aligned}\mathcal{G}_N'(1)&=m^{-N}\sum_{n=0}^{N}\binom{N}{n}(m-1)^{N-n}\Big[2(n+1)H_n-4n\Big]\\&=2m^{-N}(m-1)^{N-1}NT_{N-1}+2-2\Big(1-\frac{1}{m}\Big)^N+2m^{-N}(m-1)^NT_N-\frac{4N}{m}\\&=\frac{2N}{m}H_{N-1}-\frac{2N}{m}\sum_{k=1}^{N-1}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k+2-2\Big(1-\frac{1}{m}\Big)^N\\&+2H_N-2\sum_{k=1}^{N}\frac{1}{k}\Big(1-\frac{1}{m}\Big)^k-\frac{4N}{m}.\end{aligned}$$

Further,

$$\begin{aligned}\mathcal{G}_N''(1)=m^{-N}\sum_{n=0}^{N}\binom{N}{n}(m-1)^{N-n}\Big[&4(n+1)^2(H_n^2-H_n^{(2)})\\&-4(n+1)(4n+1)H_n+n(23n+17)\Big]\\=\frac{8N(N-1)}{m^2}\Big(1-\frac{1}{m}\Big)^{N-2}&U_{N-2}+\frac{24N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}U_{N-1}+8\Big(1-\frac{1}{m}\Big)^NU_N\\-\frac{16N(N-1)}{m^2}\Big(1-\frac{1}{m}\Big)^{N-2}&T_{N-2}-\frac{20N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}T_{N-1}+12\Big(1-\frac{1}{m}\Big)^NT_N\\-\frac{8N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}&S_{N-1}-16\Big(1-\frac{1}{m}\Big)^NS_N\\-20+\frac{16N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}&+20\Big(1-\frac{1}{m}\Big)^N+\frac{8N}{m}+\frac{23N(N-1)}{m^2}.\end{aligned}$$

One can now plug in the aforementioned expliced formulæ for S, T, U, which we don't display, because of length. Instead, we decided to produce an asymptotic formula including terms of order N or higher:

$$\begin{aligned}\mathcal{G}_N''(1)\sim\frac{8N(N-1)}{m^2}&\Big(1-\frac{1}{m}\Big)^{N-2}U_{N-2}+\frac{24N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}U_{N-1}\\-\frac{16N(N-1)}{m^2}&\Big(1-\frac{1}{m}\Big)^{N-2}T_{N-2}-\frac{20N}{m}\Big(1-\frac{1}{m}\Big)^{N-1}T_{N-1}\end{aligned}$$

$$
\begin{aligned}
&+\frac{8N}{m}+\frac{23N(N-1)}{m^2}\\
&\sim\frac{8N(N-1)}{m^2}\biggl[\frac{H_{N-2}^2-H_{N-2}^{(2)}}{2}-(\log m)H_{N-2}+\frac{m-1}{N-2}+C_1(m)-C_2(m)\biggr]\\
&+\frac{24N}{m}\biggl[\frac{H_{N-1}^2-H_{N-1}^{(2)}}{2}-(\log m)H_{N-1}+\frac{m-1}{N-1}+C_1(m)-C_2(m)\biggr]\\
&-\frac{16N(N-1)}{m^2}[H_{N-2}-\log m]-\frac{20N}{m}[H_{N-1}-\log m]+\frac{8N}{m}+\frac{23N(N-1)}{m^2}.
\end{aligned}
$$

Eventually we arrive at the last result of this paper.

Theorem 3. *The expectation and variance of the internal path length of m binary search trees of altogether N nodes, are given by*

$$
\begin{aligned}
\mathcal{E}_N&=\frac{N}{m}\Bigl[2\log\frac{N}{m}+2\gamma-\frac{4}{m}\Bigr]+O(\log N),\\
\mathcal{V}_N&=\frac{N^2}{m^2}\Bigl[7-\frac{2}{3}\pi^2+8C_1(m)-8C_2(m)-4\log^2 m\Bigr]+O(N\log N).
\end{aligned}
$$

More terms in the asymptotic expansions are easily available.

5. Conclusion

This was a first step towards the analysis of the m-model of binary search trees. Much more is known about binary search trees and could/should be lifted to that level. Just to mention something explicit, one could look at the depth of node j in an m-binary search tree of N random nodes. The average of this (in the classical case) is due to Arora and Dent [1] and is related to the number of passes that the recursive algorithm Quickselect needs to find the j-th largest element, see [5, 6].

If one wants to compute higher moments, then one needs to introduce sums like

$$\sum_{k=1}^{n}\binom{n}{k}x^k\sum_{1\le h<i<j\le k}\frac{1}{hij}$$

and similar ones.

The quantity $C_2(m)$ is related to the *dilog* function.

References

[1] S. Arora and W. Dent. Randomized binary search technique. *Communications of the ACM*, 12:77–80, 1969.

[2] J. Cichoń and W. Macyna. Approximate counters for flash memory. *17th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA), Toyama, Japan*, 2011.

[3] P. Flajolet. Approximate counting: a detailed analysis. *BIT*, 25:113–134, 1985.

[4] P. Flajolet and R. Sedgewick. *Analytic Combinatorics.* Cambridge University Press, Cambridge, 2009.

[5] C. A. R. Hoare. Find (Algorithm 65). *Communications of the ACM*, 4:321–322, 1961.

[6] P. Kirschenhofer and H. Prodinger. Comparisons in Hoare's Find algorithm. *Combinatorics, Probability, and Computing*, 7:111–120, 1998.

[7] D. E. Knuth. *The Art of Computer Programming*, volume 3: Sorting and Searching. Addison-Wesley, 1973. Second edition, 1998.

[8] D. E. Knuth. *The Art of Computer Programming*, volume 1: Fundamental Algorithms. Addison-Wesley, 1973. Third edition, 1997.

[9] H. M. Mahmoud. *Evolution of Random Search Trees.* John Wiley, New York, 1992.

Helmut Prodinger, Department of Mathematics, University of Stellenbosch, 7602 Stellenbosch, South Africa

E-mail address: hproding@sun.ac.za