Average depth in a binary search tree with repeated keys

Margaret Archibald[1] and Julien Clément[2,3]

[1] *School of Mathematics, University of the Witwatersrand, P. O. Wits, 2050 Johannesburg, South Africa,*
email: marchibald@maths.wits.ac.za.
[2]*CNRS UMR 8049, Institut Gaspard-Monge, Laboratoire d'informatique, Université de Marne-la-Vallée, France*
[3]*CNRS UMR 6072, GREYC Laboratoire d'informatique, Université de Caen, France,*
email: Julien.Clement@info.unicaen.fr

Random sequences from alphabet $\{1, \ldots, r\}$ are examined where repeated letters are allowed. Binary search trees are formed from these, and the average left-going depth of the first 1 is found. Next, the right-going depth of the first r is examined, and finally a merge (or 'shuffle') operator is used to obtain the average depth of an arbitrary node, which can be expressed in terms of the left-going and right-going depths. The variance of each of these parameters is also found.

Keywords: Binary search trees, average case analysis, repeated keys, multiset, shuffle product

1 Introduction

We examine binary search trees (BSTs) formed from sequences with equal entries. A BST is a planar tree where each node has a maximum of 2 children, which are either left or right of the parent node. BSTs are a commonly used data structure in Computer Science but are usually built from distinct entries. Here we consider a suitable definition of a BST when duplicated values are allowed: the first element in the sequence is the root of the tree and thereafter elements which are strictly less than the parent node are placed to the left (as the left child) and those greater than *or equal to* the parent node are inserted as the right child (see Fig. 1 (left)).

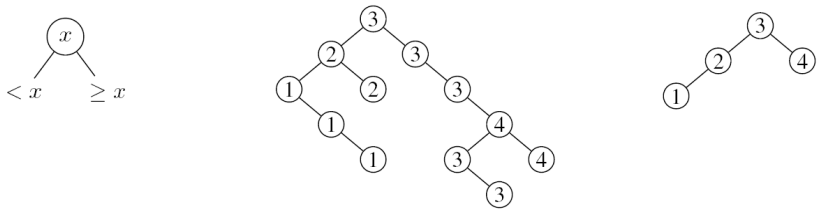

Fig. 1: The principle for binary search tree with repeated keys (left). The binary search tree of sequence 323123411343 when inserting all symbols (middle) or when inserting only the first occurrence of a symbol (right).

We examine various parameters of these trees and give an average case analysis under two standard probabilistic models ('probability' and 'multiset'). BSTs built over permutations are a very intensively studied data structure. One explanation is the close link between the construction of the tree and the *Quicksort* algorithm(i). As with many sorting algorithms, most research has been done under the assumption that all keys are distinct, i.e., that repeats are not allowed. However, given a large tree and a small pool of data from which to choose the keys, it may well happen that equal keys are common. This is a motivation for examining the case of BSTs with equal keys (see Sedgewick (1977)). Previous research on this topic includes Burge (1976), Kemp (1996) and Sedgewick (1977), where the expectation has been discussed.

Our aim in this paper is to apply modern techniques of analysis of algorithms to confirm and revisit some of these results in a somewhat simpler manner. This allows us to find both the expectation *and* the variance. Related partial results along the same lines can be found in Clément et al. (1998).

(i) The *Quicksort* algorithm runs recursively: A certain key is chosen and, by comparing it to the other keys, is placed in its final position. Thereafter, the remaining left and right subsequences (whose elements are all either greater than or less than the chosen key) are treated in the same way. For more details see Sedgewick (1977).

2 Preliminaries

We describe the situation using the same two models used in Kemp (1996) but using a symbolic approach with generating functions rather than probabilistic recurrence arguments.

2.1 Parameters on Binary search trees

The three parameters we look at in this paper are described below. We are interested in two distinct situations. In the first situation, we insert all symbols (even repeated ones) of a sequence into the binary search tree. The recursive scheme to insert a symbol x according to the value stored at the root of the tree is the following: if the tree is void, we create a new node with value x. Otherwise we go to the left or the right depending on whether x is strictly less than the value at the root or x is greater or equal (note that it is not symmetric – see Fig. 1). We examine two parameters in this setting:

- The *left-going depth of the first* 1 is the number of left-going branches from the root to the node corresponding to the first node labelled 1. This is also the longest path on the left-most side of the tree, and is numerically equal to one fewer than the number of strict left-to-right minima in the word from which the tree is built. In Figure 1, the left-going depth of the first 1 is 2, and the number of (strict) left-to-right minima in the corresponding sequence ($\mathbf{32}3\mathbf{1}23411343$) is 3.
- For an alphabet of $\{1, 2, \ldots, r\}$, finding the *right-going depth of the first node labelled* $\boldsymbol{r}$ is equivalent to finding the number of weak left-to-right maxima up to the first occurrence of r, subtract one. Note that this is not necessarily the longest path to the right as in the left-going case, since we may have repeats of the letter r which would lead to subsequent right-going branches which are not counted. For example, in Figure 1, the right-going depth of the first 4 is 3, and the number of (weak) left-to-right maxima up to the first occurrence of 4 in the sequence ($\mathbf{32}3\mathbf{1}2\mathbf{34}11343$) is 4.

There is another situation where we want to insert symbols of a sequence but we want only to store the first occurrence of a symbol. Essentially we obtain an ordinary BST built over the modified sequence where only the first occurrence is kept and all subsequent other occurrences are discarded. We will study the *average depth of any symbol* $\boldsymbol{\alpha}$ in this setting.

2.2 Random models

The 'multiset' model. For the first model we assume that the input sequences of length n are formed from the multiset $\{n_1 \cdot 1\ ;\ n_2 \cdot 2\ ;\ \ldots\ ;\ n_r \cdot r\}$. That is, we know how many times each letter i occurs in the sequence, and denote this by n_i. There are $\binom{n}{n_1,\ldots,n_r}$ possible sequences (where $n_1 + n_2 + \cdots + n_r = n$) and all are equally likely to occur. It suffices to consider the alphabet $\{1, 2, \ldots, r\}$, as we are only interested in the letters relative to each other. Any other alphabet with such an ordering would be dealt with in the same way, hence the assumption that $n_i > 0$, for $i \in \{1, \ldots, r\}$.

The 'probability' model. The second model is sometimes called the 'memoryless' model or the 'Bernoulli' model. A probability is attached to every letter in the alphabet, so the letter i would appear in the sequence with probability p_i. The sequence of length n consists of letters chosen independently from the alphabet $\{1, \ldots, r\}$. We assume that the probabilities of the letters in the alphabet add up to 1, and that each probability is non-zero. The probability distribution function is thus well-defined.

3 Results

We let n_i denote the number of times the letter i occurs in the word, and let p_i denote the probability with which the letter i occurs in the word. We use a shorthand of, for example, $N_{[i,j]} := n_i + n_{i+1} + \cdots + n_{j-1} + n_j$, or $P_{[i,i]} := p_i$. The symbols 'lg' and 'rg' represent the left-going and right-going paths, and the 'm' and 'p' denote the model used ('multiset' or 'probability').

Theorem 1 *The expected value of the left-going depth of the first 1 is (multiset model and probability model respectively)*

$$\mathbb{E}^{\mathrm{m}}_{\mathrm{lg}} = \sum_{i=2}^{r} \frac{n_i}{N_{[1,i]}} \qquad \textit{and} \qquad \mathbb{E}^{\mathrm{p}}_{\mathrm{lg}} \sim \sum_{i=2}^{r} \frac{p_i}{P_{[1,i]}}, \quad \textit{as } n \to \infty.$$

The variance in each case is

$$\mathbb{V}^{\mathrm{m}}_{\mathrm{lg}} = \sum_{i=2}^{r} \frac{n_i}{N_{[1,i]}} \left(1 - \frac{n_i}{N_{[1,i]}}\right) \qquad \textit{and} \qquad \mathbb{V}^{\mathrm{p}}_{\mathrm{lg}} \sim \sum_{i=2}^{r} \frac{p_i}{P_{[1,i]}} \left(1 - \frac{p_i}{P_{[1,i]}}\right), \quad \textit{as } n \to \infty.$$

Theorem 2 *The multiset model and probability*[(ii)] *model give the expected value of the right-going depth of the first r as*

$$\mathbb{E}^{\mathrm{m}}_{\mathrm{rg}} = \sum_{i=1}^{r-1} \frac{n_i}{N_{[i+1,r]}+1} \qquad \textit{and} \qquad \mathbb{E}^{\mathrm{p}}_{\mathrm{rg}} \sim \sum_{i=1}^{r-1} \frac{p_i}{P_{[i+1,r]}}, \quad \textit{as } n \to \infty.$$

The variances are

$$\mathbb{V}^{\mathrm{m}}_{\mathrm{rg}} = \sum_{i=1}^{r-1} \frac{n_i}{N_{[i+1,r]}+1} \left(1 - \frac{n_i}{N_{[i+1,r]}+1} + 2\frac{n_i-1}{N_{[i+1,r]}+2}\right),$$

and

$$\mathbb{V}^{\mathrm{p}}_{\mathrm{rg}} \sim \sum_{i=1}^{r-1} \frac{p_i}{P_{[i+1,r]}} \left(1 + \frac{p_i}{P_{[i+1,r]}}\right), \quad \textit{as } n \to \infty.$$

Theorem 3 *Considering a* BST *built over a sequence and where only the first occurrence of a symbol is inserted, the expected depth of some $\alpha \in \{1,\dots,r\}$ is given by*

$$\mathbb{E}^{\mathrm{m}}_{\alpha} = \sum_{i=1}^{\alpha-1} \frac{n_i}{N_{[i,\alpha]}} + \sum_{i=\alpha+1}^{r} \frac{n_i}{N_{[\alpha,i]}} \qquad \textit{and} \qquad \mathbb{E}^{\mathrm{p}}_{\alpha} \sim \sum_{i=1}^{\alpha-1} \frac{p_i}{P_{[i,\alpha]}} + \sum_{i=\alpha+1}^{r} \frac{p_i}{P_{[\alpha,i]}}, \quad \textit{as } n \to \infty.$$

and the variance of the depth of some $\alpha \in \{1,\dots,r\}$ can be expressed (by the multiset model) as

$$\mathbb{V}^{\mathrm{m}}_{\alpha} = \sum_{i=1}^{\alpha-1} \frac{n_i}{N_{[i,\alpha]}} \left(1 - \frac{n_i}{N_{[i,\alpha]}}\right) + \sum_{i=\alpha+1}^{r} \frac{n_i}{N_{[\alpha,i]}} \left(1 - \frac{n_i}{N_{[\alpha,i]}}\right) + 2\sum_{i=1}^{\alpha-1}\sum_{j=\alpha+1}^{r} \frac{n_i n_j n_\alpha}{N_{[i,j]}N_{[i,\alpha]}N_{[\alpha,j]}}.$$

Using the probability model, the variance of the depth of α as $n \to \infty$ is

$$\mathbb{V}^{\mathrm{p}}_{\alpha} \sim \sum_{i=1}^{\alpha-1} \frac{p_i}{P_{[i,\alpha]}} \left(1 - \frac{p_i}{P_{[i,\alpha]}}\right) + \sum_{i=\alpha+1}^{r} \frac{p_i}{P_{[\alpha,i]}} \left(1 - \frac{p_i}{P_{[\alpha,i]}}\right) + 2\sum_{i=1}^{\alpha-1}\sum_{j=\alpha+1}^{r} \frac{p_i p_j p_\alpha}{P_{[i,j]}P_{[i,\alpha]}P_{[\alpha,j]}}.$$

Similar patterns to these results occur in other papers in the field, for example Rivest (1976) shows that the average search time of the move-to-front heuristic is $1 + 2\sum \frac{p_i p_j}{p_i+p_j}$, where the asymptotic probability that R_i is before R_j in the list is given by $\frac{p_i}{p_i+p_j}$.

4 Methods

We will use the usual methodology of analytic combinatorics. We consider sets of words (sequences of symbols) and their corresponding generating functions to get precise average case analysis (here expected value and variance) of some parameters.

4.1 Languages and generating functions

Regular languages. A *language* L is a set of words over a fixed alphabet A. The structurally simplest (yet nontrivial) languages are the *regular languages* that can be defined in a variety of ways: by regular expressions and by finite automata. Concatenation of languages is denoted by a product ($L_1 \cdot L_2 = \{w_1 w_2 \,|\, w_1 \in L_1, w_2 \in L_2\}$). Union of languages is the ordinary set union. The empty word is denoted by ϵ and the Kleene star operator is understood as $L^* = \epsilon + L + L \cdot L + \cdots$. A regular language over an alphabet A is built by recursively applying concatenation, union and Kleene star operator to the singleton languages $\{\epsilon\}$ and $\{\sigma\}$ ($\forall \sigma \in A$). A regular expression is a description of a regular language (most commonly using symbols "$\cdot, +, *$").

(ii) Note that substituting np_i to n_i in the multiset model with p_i's being probabilities, we have consistently

$$\mathbb{E}^{\mathrm{m}}_{\mathrm{rg}} = \sum_{i=1}^{r-1} \frac{np_i}{1 + np_{i+1} + \cdots + np_r} \sim \sum_{i=1}^{r-1} \frac{p_i}{P_{[i+1,r]}}, \text{ as } n \to \infty.$$

Generating functions. The ordinary generating function (OGF) $L(x_1, \ldots, x_r)$ of $\mathcal{L}$ is, noting $|w|_i$ as the number of occurrences of the ith symbol of the alphabet A in w,

$$L(x_1, \ldots, x_r) = \sum_{w \in \mathcal{L}} x_1^{|w|_1} \ldots x_r^{|w|_r}.$$

In general, one wants to study some parameter $\gamma : A^* \to \mathbb{N}$ over some language $\mathcal{L}$. So denoting this parameter by the variable u, we can define another generating function

$$L_\gamma(u, x_1, \ldots, x_r) = \sum_{w \in \mathcal{L}} u^{\gamma(w)} x_1^{|w|_1} \ldots x_r^{|w|_r}.$$

Unambiguous regular expressions. There are fundamental links between regular languages and rational generating functions (see Flajolet and Sedgewick (2006)). A regular expression corresponding to a language $\mathcal{L}$ is said to be unambiguous if for any word in $\mathcal{L}$ the parsing of w according to the regular expression is unique. Symbols of the alphabet A and the empty word ε are of course unambiguous. In a recursive manner, considering two regular expressions e and f and their associated languages $\mathcal{L}_e$ and $\mathcal{L}_f$, then the expression $e \cdot f$ is unambiguous if e and f are unambiguous and for any word w of $\mathcal{L}_{e\cdot f} = \mathcal{L}_e \cdot \mathcal{L}_f$ there is a unique factorization $w = w_1 \cdot w_2$ and $w_1 \in \mathcal{L}_e$ and $w_2 \in \mathcal{L}_f$. This property is easily extended to the Kleene star operation (the word is uniquely parsed as a sequence). Along the same lines the regular expression $e + f$ is unambiguous if e and f are unambiguous and $\mathcal{L}_e \cap \mathcal{L}_f = \emptyset$.

When the regular expression is unambiguous, we have the simple dictionary mapping where for a regular expression e we denote by $L_e(x_1, \ldots, x_r)$ the generating function of the corresponding language:

Empty word: $\varepsilon \mapsto 1$,	Symbols: $a \in A \mapsto x_a$,	Union: $e + f \mapsto L_e + L_f$,
Catenation product: $e \cdot f \mapsto L_e \times L_f$,	Kleene star: $e^* \mapsto \frac{1}{1-L_e}$	

It is also important to note that the converse is true. Some generating functions can be thoroughly read as an unambiguous regular expression (not necessarily unique). In some cases, it can be easier to work directly with regular expressions than on generating functions as it will be illustrated in Section 5.3.

Shuffle product. We also use a well-known tool in language theory called the 'shuffle' product between two languages. By applying the shuffle product to two words we end up with all possible combinations of the original words with the letters interwoven, but with the original order within the two words unchanged. For example, take the two words ab and cd. If we shuffle these like playing cards, we get $\binom{4}{2} = 6$ solutions: $\{abcd, acbd, acdb, cabd, cadb, cdab\}$. The definition of the shuffle product is

$$au \sqcup\!\sqcup bv := a(u \sqcup\!\sqcup bv) + b(au \sqcup\!\sqcup v).$$

This definition enables us to consider the generating function of the shuffle of two languages (without parameters). When parameters (marked by a variable u) are involved the shuffle product translates also to generating functions provided the parameters considered are "compatible" with the shuffle product (meaning that for a word $w \in s \sqcup\!\sqcup t$, the parameter $\delta(w)$ can be expressed as $\delta_1(s) + \delta_2(t)$). Such a case will appear in Section 5.3.

The case where the shuffle product applies to two languages with distinct alphabets[(iii)] is of particular interest. For instance the shuffle product translates directly to generating functions in the natural product of the corresponding *exponential* generating functions (see Flajolet and Sedgewick (2006)). However only ordinary generating functions appear in this paper and we will use a clever way to compute shuffle products of ordinary generating functions (see Archibald (2005) for an alternative approach).

4.2 Expected value and variance

The 'multiset' model. One gets the average value of the parameter γ over the words in the multiset $\{n_1 \cdot 1 \ ; \ n_2 \cdot 2 \ ; \ \ldots \ ; \ n_r \cdot r\} \cap \mathcal{L}$ by considering the $x_1^{n_1} \ldots x_r^{n_r}$ coefficient in

$$\mathbb{E}_\gamma^{\mathrm{m}} = (L_\gamma)'\big|_{u=1} := \tfrac{\partial}{\partial u} L_\gamma(u, x_1, \ldots, x_r)\big|_{u=1},$$

and dividing by the number of words in $\mathcal{L}$ with symbol 1 occurring n_1 times, symbol 2 occurring n_2 times etc.. In the simplest case where $\mathcal{L}$ is the set of all words A^*, we divide through by the total number of possibilities for words of length n from the alphabet $\{1, \ldots, r\}$, i.e., $\binom{n}{n_1, \ldots, n_r}$.

[(iii)] In this paper, the alphabets involved in shuffle products will always be distinct.

When $\mathcal{L} = A^*$ one studies the variance, there is a slightly more complex formula mirroring the equality, for a random variable X, $\mathbb{V}[X] = \mathbb{E}[X(X-1)] + \mathbb{E}[X] - \mathbb{E}[X]^2$

$$\mathbb{V}^{\mathrm{m}}_{\gamma} = \frac{[x_1^{n_1}\cdots x_r^{n_r}](L_\gamma)''\big|_{u=1}}{\binom{n}{n_1,\ldots,n_r}} + \frac{[x_1^{n_1}\cdots x_r^{n_r}](L_\gamma)'\big|_{u=1}}{\binom{n}{n_1,\ldots,n_r}} - \left(\frac{[x_1^{n_1}\cdots x_r^{n_r}](L_\gamma)'\big|_{u=1}}{\binom{n}{n_1,\ldots,n_r}}\right)^2. \tag{1}$$

The 'probability' model. As in the previous model, we find the expectation and variance by partial differentiation and finding coefficients. In this case the calculations are simpler as the symbols $zp_1, \ldots, zp_r$ (where p_i is a probability) are substituted for the 'place-holder' variables $x_1, \ldots, x_r$. With the substitution $x_i \mapsto zp_i$, one has exactly

$$L^{\mathrm{p}}_{\gamma} := L_\gamma(u, zp_1, \ldots, zp_r) = \sum_{w\in\mathcal{L}} z^{|w|}u^{\gamma(w)}\mathrm{Prob}(w), \quad \text{with} \quad \mathrm{Prob}(w) = \prod_{i=1}^{|w|} p_{w_i},$$

where $\mathrm{Prob}(w)$ is the probability of w among words of same length.

The expected value for a word of length $|w| = n$ can thus be found using

$$\mathbb{E}^{\mathrm{p}}_{\gamma} = [z^n]\tfrac{\partial}{\partial u}L_\gamma(u, zp_1, \ldots, zp_r)\big|_{u=1} = [z^n](L^{\mathrm{p}}_{\gamma})'\big|_{u=1},$$

and the variance is given by

$$\mathbb{V}^{\mathrm{p}}_{\gamma} = [z^n](L^{\mathrm{p}}_{\gamma})''\big|_{u=1} + [z^n](L^{\mathrm{p}}_{\gamma})'\big|_{u=1} - \left([z^n](L^{\mathrm{p}}_{\gamma})'\big|_{u=1}\right)^2. \tag{2}$$

4.3 Extracting coefficients

In the following we will need to extract coefficients of multivariate generating functions. Below we present some lemmas on which most of the coefficient extraction machinery relies.

Lemma 1 *Suppose $n_1, n_2, \ldots, n_k \geq 0$. One has*

$$[v_1^{n_1}v_2^{n_2}\ldots v_k^{n_k}]\prod_{i=1}^{k}\frac{1}{1-(v_i+\cdots+v_k)} = \prod_{i=1}^{k}\binom{n_1+\cdots+n_i+i-1}{n_i}.$$

Proof. The z^n coefficient of an analytic function $f(z)$ is given by the Taylor expansion at $z=0$, namely

$$[z^n]f(z) = \frac{1}{n!}\tfrac{\partial^n}{\partial z^n}f(z)\big|_{z=0}.$$

The lemma can be proved by induction by examining in turn each variable $v_1, \ldots, v_n$.

Lemma 2 *Let $f(z) = \sum_{n\geq 0} f_n z^n$ be a generating function and $X = x_1 + \cdots + x_m$. Then we have*

$$[x_1^{n_1}\cdots x_m^{n_m}]f(X) = \tbinom{n_1+\cdots+n_m}{n_1,\ldots,n_m} f_{n_1+\cdots+n_m}.$$

Proof. It is exactly the definition of the multinomial coefficient $\binom{n_1+\cdots+n_m}{n_1,\ldots,n_m}$.

The last lemma proves useful to give the main term of the asymptotic expansion in the 'probability' model. Its proof relies on standard techniques in analytic combinatorics (see Flajolet and Sedgewick (2006)) and is omitted here.

Lemma 3 *Let $f(z)$ be an analytic function on $\mathbb{C}$. Suppose that f admits a simple pole at $z=1$ and that other singularities have radii greater or equal to $\rho > 1$ then the limit $\lim_{z\to 1}(1-z)f(z)$ exists and*

$$[z^n]f(z) \sim \lim_{z\to 1}(1-z)f(z), \qquad \text{as } n\to\infty.$$

5 Analysis

5.1 Left-going depth of first 1

We can express all possible words from alphabet $\{1,\dots,r\}$ symbolically as

$$\{1,\dots,r\}^* = \big(\varepsilon + r\{r\}^*\big)\big(\varepsilon + (r-1)\{r-1,r\}^*\big)\dots\big(\varepsilon + 1\{1,\dots,r\}^*\big).$$

This symbolic equation can be expressed as a generating function where u counts all (but one) of the left-to-right minima (which will correspond to the relevant left-going branches of the corresponding tree) and $x_1,\dots,x_r$ respectively mark the number of 1's,...,r's. We use the notation $X_{[i,j]} := x_i + x_{i+1} + \cdots + x_{j-1} + x_j$ to write

$$f_{\mathrm{lg}}(u,x_1,\dots,x_r) := \left(1 + \frac{x_1}{1-X_{[1,r]}}\right)\prod_{i=2}^{r}\left(1+\frac{ux_i}{1-X_{[i,r]}}\right).$$

Left-going expectation – multiset model.

For the first moment we want the partial derivative of f_{lg} with respect to u. To differentiate a sum rather than a product, we take

$$(f_{\mathrm{lg}})'\big|_{u=1} = f_{\mathrm{lg}}\big|_{u=1}\cdot\tfrac{\partial}{\partial u}\log f_{\mathrm{lg}}\big|_{u=1} = \tfrac{1}{1-X_{[1,r]}}\sum_{i=2}^{r}\tfrac{x_i}{1-X_{[i+1,r]}}. \tag{3}$$

Using first Lemma 2 and then Lemma 1 with $k=2$, $v_1 = X_{[1,i]}$ and $v_2 = X_{[i+1,r]}$ (recalling that $n = N_{[1,r]}$), we get from Equation (3)

$$[x_1^{n_1}\dots x_r^{n_r}]\tfrac{x_i}{(1-X_{[1,r]})(1-X_{[i+1,r]})} = \tfrac{n_i}{N_{[1,i]}}\tbinom{n}{n_1,\dots,n_r}.$$

Left-going expectation – probability model. In the probability model the expected value is given by substituting the probabilities zp_i in place of the formal variables x_i and extracting the z^n coefficient. So from Equation (3) and by Lemma 3 we obtain immediately (since $P_{[1,r]} = p_1 + \cdots + p_r = 1$)

$$\mathbb{E}^{\mathrm{p}}_{\mathrm{lg}} = [z^n]\frac{1}{1-z}\sum_{i=2}^{r}\frac{zp_i}{1-zP_{[i+1,r]}} \sim \sum_{i=2}^{r}\frac{p_i}{P_{[1,i]}} \text{ as } n\to\infty.$$

Note that it is easy to get the full asymptotic expansion of $\mathbb{E}^{\mathrm{p}}_{\mathrm{lg}}$ since we can determine partial fractions from Equation (3)

$$\mathbb{E}^{\mathrm{p}}_{\mathrm{lg}} = [z^n]\sum_{i=2}^{r}\frac{p_i}{P_{[1,i]}}\Big(\tfrac{1}{1-zP_{[1,r]}} - \tfrac{1}{1-zP_{[i+1,r]}}\Big) = \sum_{i=2}^{r}\frac{p_i}{P_{[1,i]}}\big(P^n_{[1,r]} - P^n_{[i+1,r]}\big) = \sum_{i=2}^{r}\frac{p_i}{P_{[1,i]}}\big(1 - P^n_{[i+1,r]}\big).$$

Left-going variance – multiset model. Using $f'' = f(\log f)'' + f'(\log f)'$, we have

$$\begin{aligned}(f_{\mathrm{lg}})''\big|_{u=1} &= \frac{1}{1-X_{[1,r]}}\left(\left(\sum_{i=2}^{r}\frac{x_i}{1-X_{[i+1,r]}}\right)^2 - \sum_{i=2}^{r}\frac{x_i^2}{(1-X_{[i+1,r]})^2}\right)\\ &= 2\sum_{i=2}^{r}\sum_{j=i+1}^{r}\frac{x_ix_j}{(1-X_{[1,r]})(1-X_{[i+1,r]})(1-X_{[j+1,r]})}.\end{aligned} \tag{4}$$

By Lemma 2 and Lemma 1 (with $k=3$, $v_1 = X_{[1,i]}$, $v_2 = X_{[i+1,j]}$ and $v_3 = X_{[j+1,r]}$) we obtain

$$[x_1^{n_1}\dots x_r^{n_r}]\tfrac{1}{1-X_{[1,r]}}\tfrac{x_i}{1-X_{[i+1,r]}}\tfrac{x_j}{1-X_{[j+1,r]}} = \tfrac{n_1n_j}{N_{[1,i]}N_{[1,j]}}\tbinom{n}{n_1,\dots,n_r}.$$

We thus have a variance of:

$$2\sum_{i=2}^{r}\sum_{j=i+1}^{r}\frac{n_in_j}{N_{[1,i]}N_{[1,j]}} + \sum_{i=2}^{r}\frac{n_i}{N_{[1,i]}} - \left(\sum_{i=2}^{r}\frac{n_i}{N_{[1,i]}}\right)^2.$$

Partial cancellation of the first and third terms simplifies the variance to the result in Theorem 1.

Left-going variance – probability model. After the substitutions $x_i \mapsto zp_i$ in Equation (4) and from Lemma 3 we get

$$[z^n](f_{\mathrm{lg}}^{\mathrm{p}})''\big|_{u=1} \sim 2\sum_{i=2}^{r}\sum_{j=i+1}^{r}\frac{p_ip_j}{(1-P_{[i+1,r]})(1-P_{[j+1,r]})} = 2\sum_{i=2}^{r}\sum_{j=i+1}^{r}\frac{p_ip_j}{P_{[1,i]}P_{[1,j]}}.$$

Again we note that we also have access to the complete asymptotic expansion with minimum work, however we chose to restrict ourselves in this paper to give only the main term for the asymptotics. We simplify the expression of the variance to yield the result of Theorem 1. ■

5.2 Right-going depth of first r

Another way to express all words from the alphabet $\{1,\dots,r\}$ symbolically is

$$\{1,\dots,r\}^* = (\varepsilon + 1\{1\}^*)(\varepsilon + 2\{1,2\}^*)(\varepsilon + 3\{1,2,3\}^*)...(\varepsilon + r\{1,\dots,r\}^*).$$

In the corresponding generating function, u will count only those nodes which will cause a right-going branch (this corresponds to the weak left-to-right maxima up to – but not including – the first occurrence of r). We have

$$f_{\mathrm{rg}}(u,x_1,\dots,x_r) := \left(1+\frac{x_r}{1-X_{[1,r]}}\right)\prod_{i=1}^{r-1}\left(1+\frac{ux_i}{1-(X_{[1,i-1]}+ux_i)}\right), \tag{5}$$

Right-going expectation – multiset model. Equation (5) leads to

$$(f_{\mathrm{rg}})'\big|_{u=1} = \frac{1}{1-X_{[1,r]}}\sum_{i=1}^{r-1}\frac{x_i}{1-X_{[1,i]}}. \tag{6}$$

Again we make use of Lemma 1 and Lemma 2, and in a similar manner as for the left-going expectation, we get (for any fixed i)

$$[x_1^{n_1}\dots x_r^{n_r}]\frac{x_i}{(1-X_{[1,i]})(1-X_{[1,r]})} = \frac{n_i}{N_{[i+1,r]}+1}\binom{n}{n_1,\dots,n_r},$$

from which the result follows.

Right-going expectation – probability model. Thanks to Lemma 3, we can make use of (6) to get

$$\mathbb{E}_{\mathrm{rg}}^{\mathrm{p}} = [z^n]\frac{1}{1-z}\sum_{i=1}^{r-1}\frac{zp_i}{1-zP_{[1,i]}} \sim \sum_{i=1}^{r-1}\frac{p_i}{P_{[i+1,r]}} \text{ as } n\to\infty.$$

Right-going variance – multiset model. Using the generating function from (5), and the expression for the variance as given in (1), we start with

$$(f_{\mathrm{rg}})''\big|_{u=1} = 2\sum_{i=1}^{r-1}\frac{x_i^2}{(1-X_{[1,r]})(1-X_{[1,i]})^2} + 2\sum_{i=1}^{r-1}\sum_{j=i+1}^{r-1}\frac{x_ix_j}{(1-X_{[1,r]})(1-X_{[1,i]})(1-X_{[1,j]})}. \tag{7}$$

Again we use Lemma 1 and Lemma 2 to extract coefficients in these expressions. For the first summand, by using the substitutions $v_1 = X_{[i+1,r]}$, $v_2 = 0$, $v_3 = X_{[1,i]}$ in Lemma 1 one has

$$[x_1^{n_1}\dots x_r^{n_r}]\frac{x_i^2}{(1-z(X_{[1,i]}+X_{[i+1,r]}))(1-zX_{[1,i]})^2} = \frac{n_i(n_i-1)}{(N_{[i+1,r]}+2)(N_{[i+1,r]}+1)}\binom{n}{n_1,\dots,n_r}.$$

On the other hand, from Lemma 1 we have (with $v_1 = X_{[j+1,r]}$, $v_2 = X_{[i+1,j]}$, $v_3 = X_{[1,i]}$)

$$[x_1^{n_1}\dots x_r^{n_r}]\frac{x_ix_j}{(1-X_{[1,r]})(1-X_{[1,j]})(1-X_{[1,i]})} = \frac{n_in_j}{(N_{[i+1,r]}+1)(N_{[j+1,r]}+1)}\binom{n}{n_1,\dots,n_r}.$$

After some cancellations, the variance as in Theorem 2 can be found.

Right-going variance – probability model. We only lack term one in (2), and for this we use (7) and apply Lemma 3. We have

$$[z^n](f_{\mathrm{rg}}^{\mathrm{p}})''\big|_{u=1} \sim 2\left(\sum_{i=1}^{r-1}\frac{p_i^2}{(1-P_{[1,i]})^2} + \sum_{i=1}^{r-1}\sum_{j=i+1}^{r-1}\frac{p_ip_j}{(1-P_{[1,i]})(1-P_{[1,j]})}\right),$$

and consequently Theorem 2 is complete. ■

5.3 Expected depth of an arbitrary node α

Introduction. The cost of searching for an arbitrary key α can be thought of as the number of comparisons in searching for a key or as the length of the path from the root to the node α, as in the previous cases.

Here we consider a different setting from the previous section in the sense that a key is inserted only once in the tree. Why is this different to the distinct key case? If only the distinct keys are allowed into the tree, the BST will always only have r nodes. However, since it was formed from a multiset of $\{1, \dots, r\}$, each tree will appear with a different probability than if it originated from a sequence with distinct keys. For example, the sequences 3321, 3231, 3221, 3213, 3212 and 3211 all produce the same tree.

We are interested in the depth of a key α in a BST built from a word $w \in A^*$. Here we implicitly suppose that α occurs in w so that we can factor w according to the first occurrence of α, i.e., one has $w = s\alpha t$ where $s \in (A \setminus \{\alpha\})^*$ and $t \in A^*$. The node in the tree (and by consequence the depth) corresponding to α only depends on s (that is on symbols prior to the first occurrence of α in w). The depth of α is one plus the sum of two quantities: the number of maximal records in $s_{<\alpha}$ and the number of minimal records in $s_{>\alpha}$, where $s_{<\alpha}$ (resp. $s_{>\alpha}$) is obtained from s by cancelling symbols greater (resp. smaller) than α. Note that this is very similar to what happens "algorithmically" when one wants to insert at the root in a BST.

One has the following generating function for the depth of a key α

$$f_\alpha(u, x_1, \dots, x_r) := \big[N_{\min}(u, x_1, \dots, x_{\alpha-1}) \text{ ш } N_{\max}(u, x_{\alpha+1}, \dots, x_r)\big] \frac{x_\alpha}{1 - X_{[1,r]}}, \tag{8}$$

Here the shuffle product takes place between OGFs $N_{\max}$ (which counts the number of left-to-right maxima in the letters smaller than α to the left of the first α) and $N_{\min}$ (which counts the number of left-to-right minima of the letters larger than α to the left of the first α). More formally, one has

$$N_{\min}(u, x_1, \dots, x_{\alpha-1}) \text{ ш } N_{\max}(u, x_{\alpha+1}, \dots, x_r) = \sum_{w \in (A\setminus\{\alpha\})^*} x_1^{|w|_1} \dots x_r^{|w|_r} u^{N_{\min}(w_{>\alpha}) + N_{\max}(w_{<\alpha})}.$$

In this instance the shuffle product is a way to generate all words w of $(A \setminus \{\alpha\})^*$, bearing in mind that w is made of two sets of letters $\{1, \dots, \alpha - 1\}$ and $\{\alpha + 1, \dots, r\}$. The factor x_α in Eq. (8) represents the first occurrence of α, and the remaining factor of $\frac{1}{1-X_{[1,r]}}$ represents everything to the right of the first α which can be of any length and which consists of any letters from 1 to r (with repeats). The variable u counts all left-to-right maxima (resp. minima).

We now define the OGFs $N_{\max}$ and $N_{\min}$. If ε represents an empty word, then the symbolic expression

$$\big(\varepsilon + 1\{1\}^*\big)\big(\varepsilon + 2\{1,2\}^*\big)\big(\varepsilon + 3\{1,2,3\}^*\big) \cdots \big(\varepsilon + (\alpha-1)\{1, \dots, \alpha-1\}^*\big)$$

translates into the generating function

$$N_{\max}(u, x_1, \dots, x_{\alpha-1}) := \prod_{i=1}^{\alpha-1} \left(1 + \frac{u x_i}{1 - X_{[1,i]}}\right).$$

Similarly, one has an expression for $N_{\min}(u, x_{\alpha+1}, \dots, x_r)$

$$N_{\min}(u, x_{\alpha+1}, \dots, x_r) := \prod_{i=\alpha+1}^{r} \left(1 + \frac{u x_i}{1 - X_{[i,r]}}\right).$$

Expectation – multiset model. We note that

$$\frac{\partial}{\partial u}(N_{\min} \text{ ш } N_{\max})\big|_{u=1} = \underbrace{\frac{\partial}{\partial u} N_{\min}\big|_{u=1} \text{ ш } N_{\max}\big|_{u=1}}_{\dagger} + \underbrace{N_{\min}\big|_{u=1} \text{ ш } \frac{\partial}{\partial u} N_{\max}\big|_{u=1}}_{\ddagger}.$$

For † we have:

$$\frac{\partial}{\partial u} N_{\min}\big|_{u=1} = \frac{\partial}{\partial u} \log N_{\min}\big|_{u=1} \cdot N_{\min}\big|_{u=1} = \frac{1}{1-X_{[\alpha+1,r]}} \sum_{i=\alpha+1}^{r} \frac{x_i}{1-X_{[i+1,r]}},$$

and

$$N_{\max}\big|_{u=1} = \prod_{i=1}^{\alpha-1} \left(1 + \frac{u x_i}{1 - X_{[1,i]}}\right)\Big|_{u=1} = \frac{1}{1 - X_{[1,\alpha-1]}}.$$

We use, when $i > \alpha$, the interpretation $\{\alpha+1,\dots,r\}^*\{i\}\{i+1,\dots,r\}^*$ for $\frac{1}{1-X_{[\alpha+1,r]}}\frac{x_i}{1-X_{[i+1,r]}}$. This is an unambiguous expression (words are decomposed with respect to the last occurrence of i). Then we shuffle this language with $\{1,\dots,\alpha-1\}^*$ corresponding to $N_{\max}\big|_{u=1} = \frac{1}{1-X_{[1,\alpha-1]}}$. This yields the unambiguous expression

$$(\{1,\dots,\alpha-1\}\cup\{\alpha+1,\dots,r\})^*\{i\}(\{1,\alpha-1\}\cup\{i+1,\dots,r\})^*.$$

Indeed the symbol i in the expression plays here the role of a separator. Shuffling is then equivalent to distributing symbols of $\{1,\dots,\alpha-1\}$ before or after this separator. The last expression is unambiguous so we get directly the generating function

$$\tfrac{\partial}{\partial u}N_{\min}\big|_{u=1} \text{ ш } N_{\max}\big|_{u=1} = \textstyle\sum_{i=\alpha+1}^{r} \frac{x_i}{(1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]})(1-(X_{[1,\alpha-1]}+X_{[i+1,r]}))}.$$

In exactly the same manner, we obtain for ‡

$$N_{\min}\big|_{u=1} \text{ ш } \tfrac{\partial}{\partial u}N_{\max}\big|_{u=1} = \textstyle\sum_{i=1}^{\alpha-1} \frac{x_i}{(1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]})(1-(X_{[1,i-1]}+X_{[\alpha+1,r]}))}.$$

We must still include the final factor $\frac{x_\alpha}{1-X_{[1,r]}}$ appearing in Eq. (8), which is independent of u, thus

$$(f_\alpha)'\big|_{u=1} = \frac{x_\alpha}{1-X_{[1,r]}}\left[\sum_{i=1}^{\alpha-1}\frac{x_i}{(1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]}))(1-(X_{[1,i-1]}+X_{[\alpha+1,r]}))} + \sum_{i=\alpha+1}^{r}\frac{x_i}{(1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]}))(1-(X_{[1,\alpha-1]}+X_{[i+1,r]}))}\right].$$

Once again it is a direct application of Lemmas 1 and 2 to get the first expression of Theorem 3.

Expectation – probability model. In this case applying the same techniques (substitution, Lemma 3 and the fact that $\sum p_i = 1$), the expected value is that of Theorem 3, namely

$$\mathbb{E}^{\mathrm{p}}_\alpha = [z^n](f^{\mathrm{m}}_\alpha)'\big|_{u=1} \sim \sum_{i=1}^{\alpha-1}\frac{p_i}{P_{[i,\alpha]}} + \sum_{i=\alpha+1}^{r}\frac{p_i}{P_{[\alpha,i]}}, \quad \text{as } n\to\infty.$$

Variance – multiset model. We start with the generating function in (8). To find the variance we use (1), and consider

$$\tfrac{\partial^2}{\partial u^2}(N_{\max} \text{ ш } N_{\min})\big|_{u=1} = \underbrace{\tfrac{\partial^2}{\partial u^2}N_{\max}\big|_{u=1} \text{ ш } N_{\min}\big|_{u=1}}_{\sharp} + 2\underbrace{\tfrac{\partial}{\partial u}N_{\max}\big|_{u=1} \text{ ш } \tfrac{\partial}{\partial u}N_{\min}\big|_{u=1}}_{\natural} + \underbrace{\tfrac{\partial^2}{\partial u^2}N_{\min}\big|_{u=1} \text{ ш } N_{\max}\big|_{u=1}}_{\flat}.$$

Of the above, only the two second-order partial derivatives have not been already calculated. We look at these now (recall that $f'' = f(\log f)'' + f'(\log f)'$):

$$\tfrac{\partial^2}{\partial u^2}N_{\max}\big|_{u=1} = \frac{1}{1-X_{[1,\alpha-1]}}\left(\Big(\sum_{i=1}^{\alpha-1}\frac{x_i}{1-X_{[1,i-1]}}\Big)^2 - \sum_{i=1}^{\alpha-1}\frac{x_i^2}{(1-X_{[1,i-1]})^2}\right).$$

After cancellation,

$$\tfrac{\partial^2}{\partial u^2}N_{\max}\big|_{u=1} = 2\textstyle\sum_{i=1}^{\alpha-1}\sum_{k=1}^{i-1}\frac{x_i x_k}{(1-X_{[1,\alpha-1]})(1-X_{[1,i-1]})(1-X_{[1,k-1]})}, \tag{9}$$

Similarly, one has

$$\tfrac{\partial^2}{\partial u^2}N_{\min}\big|_{u=1} = 2\textstyle\sum_{i=\alpha+1}^{r}\sum_{k=\alpha+1}^{i-1}\frac{x_i x_k}{(1-X_{[\alpha+1,r]})(1-X_{[i+1,r]})(1-X_{[k+1,r]})}. \tag{10}$$

We now proceed to the shuffle products. We recall that we want to find unambiguous regular expressions corresponding to each term of the shuffle product. Then we find an unambiguous expression for the result of this product. This gives in turn the corresponding generating function. For instance, in Eq. (9), a term is mapped to the unambiguous expression (with $k < i < \alpha$)

$$\{1,\dots,\alpha-1\}^*\{i\}\{1,\dots,i-1\}^*\{k\}\{1,\dots,k-1\}^*.$$

Here again the symbols i and k act as "separators" (or "markers"). When shuffling this language with $\{\alpha+1,\dots,r\}^* \equiv N_{\min}\big|_{u=1} = \frac{1}{1-X_{[\alpha+1,r]}}$ one gets

$$(\{1,\dots,\alpha-1\}\cup\{\alpha+1,\dots,r\})^*\{i\}(\{1,\dots,i-1\}\cup\{\alpha+1,\dots,r\})^*\{k\}(\{1,\dots,k-1\}\cup\{\alpha+1,\dots,r\})^*.$$

Going back to generating functions, we have

$$\frac{\partial^2}{\partial u^2}N_{\max}\big|_{u=1} \text{ ш } N_{\min}\big|_{u=1} = 2\frac{1}{1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]})}\times \sum_{i=1}^{\alpha-1}\sum_{k=1}^{i-1}\frac{x_i x_k}{(1-(X_{[1,i-1]}+X_{[\alpha+1,r]}))(1-(X_{[1,k-1]}+X_{[\alpha+1,r]}))},$$

Proceeding in exactly the same manner we obtain for Equation (10)

$$\frac{\partial^2}{\partial u^2}N_{\max}\big|_{u=1} \text{ ш } N_{\min}\big|_{u=1} = 2\frac{1}{1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]})}\times \sum_{i=\alpha+1}^{r}\sum_{k=\alpha+1}^{i-1}\frac{x_i x_k}{(1-(X_{[i+1,r]}+X_{[1,\alpha-1]}))(1-(X_{[k+1,r]}+X_{[1,\alpha-1]}))}.$$

Lastly, expression ♮ involves

$$\frac{\partial}{\partial u}N_{\max}\big|_{u=1} \text{ ш } \frac{\partial}{\partial u}N_{\min}\big|_{u=1} = \left(\frac{1}{1-X_{[1,\alpha-1]}}\sum_{i=1}^{\alpha-1}\frac{x_i}{1-X_{[1,i-1]}}\right) \text{ ш } \left(\frac{1}{1-X_{[\alpha+1,r]}}\sum_{i=\alpha+1}^{r}\frac{x_i}{1-X_{[i+1,r]}}\right).$$

A typical term of this sum is for $1 \le i < \alpha < j \le r$ $\frac{1}{1-X_{[1,\alpha-1]}}\frac{x_i}{1-X_{[1,i-1]}}$ ш $\frac{1}{1-X_{[\alpha+1,r]}}\frac{x_j}{1-X_{[j+1,r]}}$. So we consider

$$\{1,\dots,\alpha-1\}^*\{i\}\{1,\dots,i-1\}^* \text{ ш } \{\alpha+1,\dots,r\}^*\{j\}\{j+1,\dots,r\}^*.$$

We obtain the resulting expression, with $A=\{1,\dots,r\}$ and $1 \le i < \alpha < j \le r$ (note that the set subtract operation is solely used to shorten the length of the expressions involved)

$$(A\setminus\{\alpha\})^*\,\{i\}\,(A\setminus\{i,\dots,\alpha\})^*\,\{j\}\,(A\setminus\{i,\dots,j\})^* \bigcup\ (A\setminus\{\alpha\})^*\,\{j\}\,(A\setminus\{\alpha,\dots,j\})^*\,\{i\}\,(A\setminus\{i,\dots,j\})^*.$$

This yields

$$\frac{\partial}{\partial u}N_{\max}\big|_{u=1} \text{ ш } \frac{\partial}{\partial u}N_{\min}\big|_{u=1} = \sum_{i=1}^{\alpha-1}\sum_{j=\alpha+1}^{r}\frac{x_i x_j}{(1-(X_{[1,\alpha-1]}+X_{[\alpha+1,r]}))(1-(X_{[1,i-1]}+X_{[j+1,r]}))}\times \left(\frac{1}{1-(X_{[1,\alpha-1]}+X_{[j+1,r]})}+\frac{1}{1-(X_{[1,i-1]}+X_{[\alpha+1,r]})}\right).$$

Finally, we multiply ♯, ♭ and ♮ by $\frac{x_\alpha}{1-X_{[1,r]}}$. It remains to extract coefficients using Lemmas 1 and 2 to get $\frac{1}{\binom{n}{n_1,\dots,n_r}}[x_1^{n_1}\dots x_r^{n_r}](f_\alpha)''\big|_{u=1}$, which is

$$2\left(\sum_{i=1}^{\alpha-1}\sum_{j=1}^{i-1}\frac{n_i n_j}{N_{[j,\alpha]}N_{[i,\alpha]}}+\sum_{i=\alpha+1}^{r}\sum_{j=i+1}^{r}\frac{n_i n_j}{N_{[\alpha,i]}N_{[\alpha,j]}}+\sum_{i=1}^{\alpha-1}\sum_{j=\alpha+1}^{r}\frac{n_i n_j}{N_{[i,j]}}\left(\frac{1}{N_{[i,\alpha]}}+\frac{1}{N_{[\alpha,j]}}\right)\right).$$

Some simplification completes the third proof in Theorem 3.

Variance – probability model. Using lemma 3 for $[z^n](f_\alpha^{\mathrm{p}})''\big|_{u=1}$ gives

$$[z^n](f_\alpha^{\mathrm{p}})''\big|_{u=1} \sim 2\left(\sum_{i=1}^{\alpha-1}\sum_{j=1}^{i-1}\frac{p_i p_j}{P_{[j,\alpha]}P_{[i,\alpha]}}+\sum_{i=\alpha+1}^{r}\sum_{j=i+1}^{r}\frac{p_i p_j}{P_{[\alpha,i]}P_{[\alpha,j]}}+\sum_{i=1}^{\alpha-1}\sum_{j=\alpha+1}^{r}\frac{p_i p_j}{P_{[i,j]}}\left(\frac{1}{P_{[i,\alpha]}}+\frac{1}{P_{[\alpha,j]}}\right)\right).$$

Theorem 3 is thus proved. ■

Acknowledgements

The authors wish to thank A. Lascoux, Ph. Flajolet and B. Vallée for helpful discussions and referees for comments which improved the presentation.

References

M. Archibald. *Combinatorial Problems Related to Sequences with Repeated Entries*. PhD thesis, School of Mathematics, University of the Witwatersrand, Johannesburg, South Africa, November 2005.

W. Burge. An analysis of binary search trees formed from sequences of nondistinct keys. *Journal of the Association for Computing Machinary*, 23(3):451–454, 1976.

J. Clément, P. Flajolet, and B. Vallée. The analysis of hybrid trie structures. In *Proceedings of the Ninth Annual ACM–SIAM Symposium on Discrete Algorithms*, pages 531–539, Philadelphia, 1998. SIAM Press.

P. Flajolet and R. Sedgewick. Analytic combinatorics. http://algo.inria.fr/flajolet/Publications/books.html, January 2006.

R. Kemp. Binary search trees constructed from nondistinct keys with/without specified probabilities. *Theoretical Computer Science*, 156:39–70, 1996.

R. Rivest. On self-organizing sequential search heuristics. *Communications of the ACM*, 19(2):63–67, 1976.

R. Sedgewick. Quicksort with equal keys. *SIAM Journal on Computing*, 6(2):240–267, 1977.