A Bijection for the Evolution of B-Trees

Fabian Burghart
Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Stephan Wagner
Institute of Discrete Mathematics, TU Graz, Austria
Department of Mathematics, Uppsala University, Sweden

Abstract

A B-tree is a type of search tree where every node (except possibly for the root) contains between m and $2m$ keys for some positive integer m, and all leaves have the same distance to the root. We study sequences of B-trees that can arise from successively inserting keys, and in particular present a bijection between such sequences (which we call histories) and a special type of increasing trees. We describe the set of permutations for the keys that belong to a given history, and also show how to use this bijection to analyse statistics associated with B-trees.

2012 ACM Subject Classification Mathematics of computing → Enumeration; Theory of computation → Randomness, geometry and discrete structures; Theory of computation → Data structures design and analysis

Keywords and phrases B-trees, histories, increasing trees, bijection, asymptotic enumeration, tree statistics

Digital Object Identifier 10.4230/LIPIcs.AofA.2024.10

Funding *Fabian Burghart*: F. Burghart has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Grant Agreement No 101034253.
Stephan Wagner: Supported by the Swedish research council (VR), grant 2022-04030.

1 Introduction and main results

B-trees, since their inception in [4], have become a popular data structure. Regarding their mathematical analysis, there were some early results by Yao [13] and Odlyzko [11] for the special case of 2-3-trees, but despite Knuth posing a natural open question in [7], progress has been scarce. Perhaps most notable is the approach using Pólya urns as in [1–3, 6], which yielded results especially for the fringe analysis of B-trees. In this paper, we propose a novel way of investigating B-trees, by focussing on what we call histories.

1.1 B-trees and their insertion algorithm

By a search tree, we mean a rooted plane tree whose nodes contain *keys*, which we think of as pairwise distinct real numbers, in such a way that (1) the keys are stored in increasing order from left to right (including within a single node), and (2) every non-leaf node containing k keys has exactly $k+1$ children, where we think of the i-th child as being attached between the $(i-1)$-th and the i-th key of its parent node. For $i = 1$ we interpret this as being attached to the left of the first key, and analogously for $i = k+1$, the child is attached to the right of the last key in the node. Note that we explicitly allow leaves to contain keys, and will refer to the intervals between consecutive keys in a leaf as *gaps*; thus we do not follow the convention of [10] where the leaves really take the place of our gaps, and therefore cannot contain keys.

35th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (AofA 2024).
Editors: Cécile Mailler and Sebastian Wild; Article No. 10; pp. 10:1–10:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Let $m \geq 1$. Following Knuth [10, Section 6.2.4], a *B-tree* of order $2m+1$ is a search tree satisfying the following properties: Every node contains at least m and at most $2m$ keys, except for the root which contains at least 1 and at most $2m$ keys. Moreover, the tree is balanced in the sense that all leaves have the same distance to the root. We remark that some authors (e.g. [4]) refer to such a tree as a B-tree of order m instead.

B-trees can be constructed via the following *insertion algorithm*: Given a B-tree and a key that is not already stored in the tree, place the key in the appropriate leaf and the appropriate position within the keys of the leaf. If, after this placement, the leaf still contains at most $2m$ keys, then we are done. Otherwise, we split the node containing $2m+1$ keys by moving the median key up into the parent node and grouping the lowest m keys and the largest m keys each in their own node. By doing this, it might now happen that the parent node contains $2m+1$ keys, in which case we again split it into two nodes of m vertices and move the median key (of the parent node) up. This process may propagate all the way along the path from the leaf where we inserted the key to the root vertex, in which case we create a new root vertex above the old root, containing only a single key (the one that was the median among the $2m+1$ keys of the old root), and split the old root in two. Note that the latter case of splitting the root is the only situation in which the height of the B-tree can increase.

For the purpose of this article, we are interested in B-trees up to isomorphism of rooted plane trees. Equivalently, we can represent an isomorphism class by replacing all keys by dots, as in Figure 1(left). For brevity's sake, we will henceforth use B-tree to mean such an isomorphism class. An alternative way to think about these isomorphism classes is to fix the keys instead, e.g. by saying the keys are the set $\{1, \ldots, n\}$ – the disadvantage of this approach being that inserting another key means having to reassign the values of some of the old keys. Nonetheless, we will make use of both of these representations.

1.2 Main results

Let T_n be a B-tree of order $2m+1$ containing n keys. A *history* of T_n is a finite sequence $(T_1, \ldots, T_n)$ of B-trees of order $2m+1$ such that for all $i = 2, \ldots, n$, the tree T_i is obtained from T_{i-1} through inserting a single key using the insertion algorithm outlined above. In particular, T_i contains i keys. We denote by $\mathscr{H}_m(T_n)$ the set of all histories of T_n, and by $\mathscr{H}_m(n)$ the set of all histories of any B-tree of order $2m+1$ with n keys. In other words, $\mathscr{H}_m(n) = \bigcup_{T_n} \mathscr{H}_m(T_n)$, where the union is taken over all (non-isomorphic) B-trees of order $2m+1$ with n keys.

We can now state our main result:

▶ **Theorem 1.** *Let $n, m \geq 1$. There is a bijection between $\mathscr{H}_m(n)$ and the set of all trees H_n satisfying the following properties:*

(i) *H_n is a rooted plane tree on n vertices, labelled by $\{1, \ldots, n\}$, such that along each path from the root to a leaf, the labels are increasing.*

(ii) *The vertices of H_n at heights $2m, 3m+1, 4m+2, \ldots$ have up to two children, all other vertices have at most one child.*

We will call trees H_n satisfying properties (i) and (ii) in the theorem $(2m+1)$-*historic* (or just *historic*, if it is not ambiguous) in the interest of brevity. Given a historic tree H on n vertices, it will be useful throughout to consider all potential positions for attaching a vertex $n+1$ that lead to another historic tree. We think of these positions as *external vertices*, and call the vertices in H *internal* to tell them apart. We also write $\bar{H}$ to denote H together with the external vertices. Furthermore, we call the internal vertices at height $2m, 3m+1, 4m+2, \ldots$ *branchings* (irrespective of how many internal children they have).

▶ **Proposition 2.** *Let H_n be the historic tree corresponding to a history $(T_1, \dots, T_n)$ of B-trees of order $2m+1$ under the bijection in Theorem 1. Then, the following holds:*

(i) *For any $n \geq 1$, the number of external vertices of H_n equals the number of leaves of T_n.*

(ii) *For any $n \geq 1$, the number of branchings in H_n equals the number of keys in T_n that are not stored in leaves.*

(iii) *Let $n \geq 2m+1$. Consider the i-th external vertex v of H_n from the left, and let s be the number of internal vertices in H_n strictly between v and the closest branching above v. Then, the i-th leaf of T_n from the left contains exactly $m+s$ keys.*

We dedicate Section 2 to the proof of Theorem 1 and Proposition 2. That section will also contain the description of the bijection. In Section 3, we exhibit a recursive construction of $\underline{\pi}(H_n)$, the set of all permutations $\pi \in S_n$ that, when used as key sequence for a B-tree, lead to the history described by the historic tree H_n. As part of this description, we obtain the following result:

▶ **Proposition 3.** *Let H_n be a $(2m+1)$-historic tree having $b \geq 1$ branchings. Let $s_1, \dots, s_{b+1}$ be the number of internal vertices in H_n strictly between the i-th external vertex and its closest branching. Then*

$$|\underline{\pi}(H_n)| = \left(\frac{(2m+1)!}{(m!)^2}\right)^b \cdot \prod_{i=1}^{b+1}(m+s_i)!. \tag{1}$$

This formula is somewhat reminiscent of the classical hook length formula, see e.g. [10, Section 5.1.4, Exercise 20]: the number of increasing labellings of a tree with n vertices is given by

$$n! \prod_v \frac{1}{N_v},$$

where the product is over all vertices and N_v is the number of vertices in the subtree consisting of v and all its descendants.

▶ Remark 4. It is possible to consider B-trees of order $2m$ as well, where a node splits whenever it is assigned $2m$ keys. In that case, the smallest $m-1$ keys end up in the left node, the m-th key is pushed into the parent node, and the largest m keys end up in the right node. It is still possible to define suitable $(2m)$-historic trees, but the distance between a branching and the next branching below will depend on whether we go to the left or to the right in H_n.

2 The bijection

The purpose of this section is to prove Theorem 1.

We begin by describing the bijection; see Figure 1 for an example. If $n = 1$, there is only one B-tree and only one corresponding H_1. For an arbitrary history $(T_1, \dots, T_n) \in \mathscr{H}_m(n)$, construct the corresponding H_n as follows: Assume we already constructed H_k corresponding to the history $(T_1, \dots, T_k) \in \mathscr{H}_m(k)$ for some $1 \leq k < n$. Then T_{k+1} is obtained from T_k by inserting a single new key. If this insertion takes place in the i-th leaf (counted from left to right) of T_k before accounting for possible splits, then we attach the vertex labelled $k+1$ to H_k at the i-th external vertex of H_k (counted from left to right). This gives H_{k+1}, and inductively, H_n.

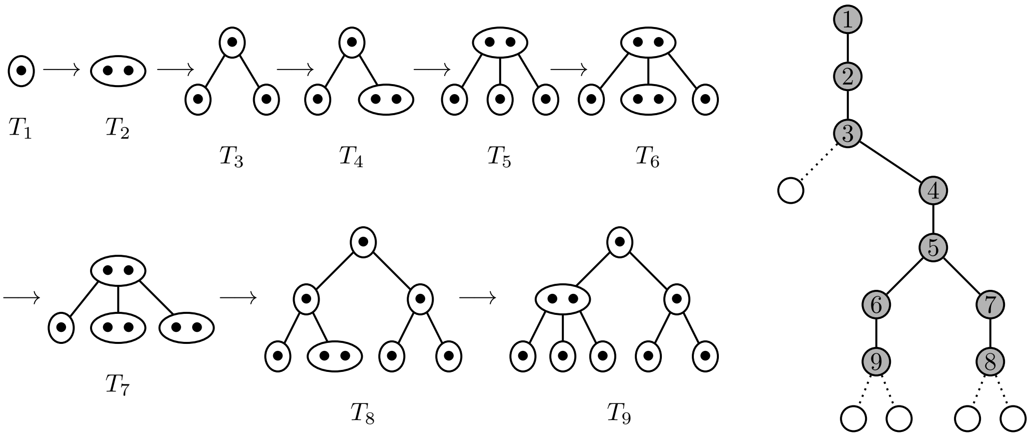


Figure 1 A history of B-trees of order $2m+1=3$ on the left, with the corresponding historic tree H_9 shown on the right. The external vertices of H_9 are shown in white, and are connected by dotted lines. The vertices $3, 5, 8$, and 9 are the branchings of H_9.

Conversely, given some historic H_n, we can construct trees $H_1, \dots, H_{n-1}$ such that H_k is the subtree consisting of the vertices with label $\leq k$. Suppose that the vertex $k+1$ is attached to $H_k \subseteq H_{k+1}$ in the i-th external vertex of H_k, and suppose we have already constructed the history $(T_1, \dots, T_k)$ corresponding to H_k. Then we can extend this history to the one corresponding to H_{k+1} by inserting a key into the i-th leaf from the left of T_k, and let T_{k+1} be the B-tree obtained by this (possibly after performing the necessary splits).

It is clear from the description that this gives inverse maps between $\mathscr{H}_m(n)$ and $(2m+1)$-historic trees on n vertices, provided the constructions are at all well-defined. This is the case if the number of external vertices on H_k equals the number of leaves of T_k, which is exactly claim (i) in Proposition 2. Thus we proceed by proving Proposition 2, which will imply Theorem 1.

Proof of Proposition 2. We first note that (i) is equivalent to (ii). Indeed, since all non-branchings have outdegree 1 in $\overline{H_n}$, and the branchings have outdegree exactly 2, the number of branchings is one less than the number of external vertices. Similarly, it is a simple consequence of the insertion algorithm for B-trees that every key that gets moved out of a leaf by a split increases the number of leaves by one, so that the number of keys not stored in leaves is one less than the number of leaves.

Next, we observe that (i) holds for $n \leq 2m$. This is the case since any B-tree of order $2m+1$ for those values of n only has a single node (which is simultaneously root and leaf), and all vertices with these labels in H_n necessarily have outdegree 1. We proceed by induction on n.

For $n = 2m+1$, we see the first split in T_n, leading to a root node containing a single key, and two leaves containing m keys each. For H_n, we have now reached height $2m$, and thus have two external vertices – these are the children of a branching in $\overline{H_n}$, thus there are no internal vertices between them and the branching. This establishes both (i) and (iii) for $n = 2m+1$.

Now assume that (i) and (iii) hold for some $n \geq 2m+1$, and that we obtain T_{n+1} from T_n by adding a key to the i-th leaf, which held $m+s$ keys in T_n. We distinguish two cases for s:

For $0 \leq s \leq m-1$, we end up with $m+s+1$ keys in the i-th leaf of T_{n+1}, and the number of leaves does not change. For H_n, we need to append the vertex labelled $n+1$ in place of the i-th external vertex. Denote by w the closest branching to $n+1$ (i.e., the most recent predecessor that is a branching; such a vertex exists since $n \geq 2m+1$). By assumption, there are exactly s vertices strictly between w and $n+1$, so $n+1$ is not a branching, and only has a single external child (and the path from that external vertex to w is one vertex longer). Thus, properties (i) and (iii) hold for $n+1$.

If, on the other hand, $s = m$, then adding the key splits the i-th leaf; producing two leaves in its stead that each hold m keys. For H_n, we denote again by w the closest branching to the i-th external vertex which becomes the position of the new vertex $n+1$. Invoking (iii) for H_n shows that there are m vertices between $n+1$ and w, so $n+1$ is another branching and therefore has two new external vertices as children, replacing the old one. The closest branching to the new external vertices is now $n+1$, and there are no internal vertices strictly between them, which again corresponds to the number of keys in the new vertices. This shows that splits in the B-tree correspond to branchings in the historic tree and asserts (i) and (iii) for $n+1$, finishing the induction argument. ◀

3 The permutations associated with a history

Let T be a B-tree of order $2m+1$, containing n keys. We denote by $\underline{\pi}(T)$ the set of all permutations $\pi \in S_n$ that, when used as a key sequence for the insertion algorithm, yield the tree T. The aim of this section is to give a recursive description of $\underline{\pi}(T)$ in terms of the "trimmed" tree $T^{(1)}$ that is obtained from T by deleting all leaves. For this purpose, write n_1 for the number of keys stored in $T^{(1)}$. We will rely on the following observation:

Consider a history $(T_1, \dots, T_n = T)$. Let $i_1 < i_2 < \cdots < i_{n_1}$ be those i where T_i was obtained from T_{i-1} by inserting a key that led to a split (note that this is consistent with the indexing). Then $\left(T_{i_1}^{(1)}, \dots, T_{i_{n_1}}^{(1)}\right)$ is a valid history of $T^{(1)}$. We remark that this is a purely combinatorial statement: If we instead looked at i.i.d. keys sampled from some continuous probability distribution, then the processes $(T_n)_{n\geq 1}$ and $\left(T_{i_n}^{(1)}\right)_{n\geq 1}$ would be quite different!

As a consequence, suppose we are given a permutation $\pi \in \underline{\pi}(T)$. This π produces a history $(T_1, \dots, T_n)$. Moreover, keeping track of the actual keys, we obtain a sequence $K_{i_1}, \dots, K_{i_{n_1}}$ of those keys that ascend above the leaves at times $i_1, \dots, i_{n_1}$. Forgetting about their actual values and only keeping track of the relative size of the K_{i_j} then produces a new permutation $\pi^{(1)} \in \underline{\pi}(T^{(1)})$, where, moreover, $\pi^{(1)}$ produces the history $\left(T_{i_1}^{(1)}, \dots, T_{i_{n_1}}^{(1)}\right)$. This defines a map $\Psi_T : \underline{\pi}(T) \to \underline{\pi}(T^{(1)}), \pi \mapsto \pi^{(1)}$, and our goal will be to invert this: Given a $\pi^{(1)}$, we want to find all $\pi \in S_n$ that lead to such $\pi^{(1)}$.

This inversion will come in the form of a 3-step algorithm, described in detail below, in Section 3.2. However, we will give a high-level overview now:

1. In the first step, we start from a given $\pi^{(1)} \in \underline{\pi}(T^{(1)})$, and lift it to a sequence $(K_{i_1}, \dots, K_{i_{n_1}})$ as above.
2. In the second step, we use $\pi^{(1)}$ and Proposition 2 to construct an acyclic digraph $G = G(T, \pi^{(1)})$. Lemma 7 states that the set of topological labellings of G corresponds bijectively to the set of historic trees of T that produce $\pi^{(1)}$.
3. Therefore, in the third step, we can fix a historic tree H obtained from step 2, and restrict our attention to $\underline{\pi}(H)$. The algorithm given will produce an arbitrary element of $\underline{\pi}(H)$ after making a sequence of choices; different choices will lead to different permutations, and going over all permitted choices produces the entire set, see Lemma 8. In more concrete terms, we start step 3 with an "empty" permutation consisting of n blank symbols, and by recursively comparing it against $K_{i_1}, \dots, K_{i_{n_1}}$ and H we will replace the blanks by entries from $\{1, \dots, n\}$.

3.1 Preparatory lemmas

To ensure well-definedness at a later point (Lemma 7), we need the following lemma:

▶ **Lemma 5.** *There is a well-defined map $\hat{\Psi}_T : \mathscr{H}_m(T) \to \underline{\pi}(T^{(1)})$ that assigns to a history $(T_1, \dots, T_n = T)$ the $\pi^{(1)}$ constructed above, where $\pi \in \underline{\pi}(T)$ is any permutation producing the history.*

Proof. We will assume that the keys in T_j are exactly $1, \dots, j$ (labelled from left to right, since T_j is a search tree) for $1 \le j \le n$, and relabel them accordingly whenever we insert a new key. We show inductively that we can (a) determine uniquely which key moved up from the leaves at the times $i_1, i_2, \dots$ and (b) keep track of how the keys in $T_{j-1}^{(1)}$ change as we go to $T_j^{(1)}$. For $T_1, \dots, T_{2m}$, there is nothing to show. In T_{2m+1}, we know that the unique key in the root node has label $m+1$. Suppose we have verified (a) and (b) for some $j \ge 2m+1$. If $j+1$ is not one of $i_1, i_2, \dots, i_{n_1}$ then no splits happen, and comparing T_{j+1} with T_j reveals which leaf grew by one. All keys in $T_j^{(1)}$ to the right of that leaf are increased by 1 for $T_{j+1}^{(1)}$, all other keys in $T_j^{(1)}$ remain the same. If on the other hand $j+1$ is one of $i_1, i_2, \dots, i_{n_1}$ then comparing T_{j+1} and T_j reveals which leaf of T_j split. As before, all keys in $T_j^{(1)}$ to the right of that leaf are increased by 1 for $T_{j+1}^{(1)}$, all other keys in $T_j^{(1)}$ remain the same. Moreover, let K be the largest key in $T_j^{(1)}$ to the left of the splitting leaf. Then the new key introduced to $T_{j+1}^{(1)}$ will be $K+m+1$, and it will be the unique key in T_{j+1} that is placed between the two leaves coming from the split leaf.

Thus, only from the history of T_n we can keep track of which keys were introduced to $T_j^{(1)}$ in which order, which yields $K_{i_1}, \dots, K_{i_{n_1}}$ after updating all the keys and thus $\pi^{(1)}$. ◀

We also note the following simple fact about the bijection from Theorem 1:

▶ **Lemma 6.** *Let H_n be the historic tree for $(T_1, \dots, T_n)$. Suppose that vertex i of a historic tree H is a branching, and suppose that the key that is pushed upwards from the splitting leaf at that time is $K_i \in \{1, \dots, n\}$ in T_n. Let $j \in \{i+1, \dots, n\}$ be another vertex of H, and let k_j denote the key added at time j in the history. Then, $k_j > K_i$ if and only if j is positioned to the right of i in H (not necessarily as a descendant of i), and $k_j < K_i$ otherwise. Moreover, if $j > i$ is another branching of H, then also $K_j > K_i$ if and only if j is to the right of i, and $K_j < K_i$ otherwise.*

Proof. This follows from the observation that after the split pushes K_i upwards, the leaves of the B-tree can be partitioned into those containing keys $< K_i$, which are therefore to the left, and those containing keys $> K_i$, which are further to the right, as well as from the description of the bijection given in Section 2. ◀

3.2 The algorithm

We now turn our attention to the promised "inverse" of Ψ_T. Denote by h the height of T.

For $h = 0$, the tree T only consists of the root node, and then $\underline{\pi}(T) = S_n$. For $h > 0$, suppose we know $\underline{\pi}(T^{(1)})$.

Step 1. By performing an in-order traversal of the keys in T, we can see which of the numbers $1, \dots, n$ correspond to the keys in $T^{(1)}$. In other words, in-order traversal gives a monotone injection $\iota : \{1, \dots, n_1\} \hookrightarrow \{1, \dots, n\}$, by sending i to the j that is the i-th key from the left among those not in a leaf node of T. This injection in turn allows us to write any $\pi^{(1)} \in \underline{\pi}(T^{(1)})$ as a sequence $\pi_\iota = \big(K_{i_1}, \dots, K_{i_{n_1}}\big)$.

Step 2. We construct a rooted digraph $G = G(T, \pi^{(1)})$ in the following way: First, construct a binary search tree from $\pi^{(1)}$. Then, subdivide the edges (and move the root up) in such a fashion that the nodes of the binary search tree become the branchings of a historic tree and append extra vertices to match with the leaves of T, according to Proposition 2(iii). We then direct all edges away from the root, and consider the directed path $\pi^{(1)}(1) \longrightarrow \cdots \longrightarrow \pi^{(1)}(n_1)$. Merge this path into the (mostly empty) historic tree by identifying the vertex $\pi^{(1)}(i)$ in the path with the vertex containing $\pi^{(1)}(i)$ in the tree, for all $i = 1, \ldots, n_1$. For bookkeeping, we colour the edges coming from the path red, and the edges from the tree black. Finally, delete all labels/keys from the resulting digraph G.

▶ **Lemma 7.** *The digraph $G = G(T, \pi^{(1)})$ constructed in this fashion is acyclic. Furthermore, any topological labelling of G (that is, any labelling such that all edges point towards the higher label) induces a historic tree H for T on the black edges. Such H corresponds to those histories of T that are obtained by $\pi \in S_n$ such that $\pi^{(1)}$ is the associated history of $T^{(1)}$. In other words, we have*

$$\{\textit{topological labellings of } G(T, \pi^{(1)})\} \overset{1:1}{\longleftrightarrow} \hat{\Psi}^{-1}(\pi^{(1)}),$$

where the bijection is the one from Theorem 1 after removing the red edges from G.

Step 3. It remains to give the actual description of $\underline{\pi}(T)$. Specifically, writing $\underline{\pi}(H)$ for the set of $\pi \in S_n$ that produce the history encoded by the historic tree H, we can pick an H coming from a topological labelling of $G(T, \pi^{(1)})$ and describe the corresponding $\underline{\pi}(H)$. We are given the key sequence $\pi_\iota = (K_{i_1}, \ldots, K_{i_{n_1}})$ from Step 1, as well as a fixed topological labelling of G, with the induced historic tree H.

In what follows, p will be a sequence of distinct integers which is to be determined, thought of as a map onto some range $\mathcal{R}$. Furthermore $\mathcal{H}$ will be a historic tree on $|\mathcal{R}|$ vertices, and $\mathcal{K}$ is a subsequence of π_ι containing (in the same order) all those K_{i_j} that appear in $\mathcal{R}$. Moreover, we demand that the length of the sequence $\mathcal{K}$ equals the number of branchings in $\mathcal{H}$. Then, the following recursive procedure constructs all desired π:

Step 3.0. Initialize $p = \pi$ as a yet undetermined permutation in S_n, thus $\mathcal{R} = \{1, \ldots, n\}$. Further, set $\mathcal{H} = H$ and $\mathcal{K} = \pi_\iota$.

Step 3.1. If $|\mathcal{R}| \leq 2m$, let p be an arbitrary bijection onto $\mathcal{R}$. Otherwise, choose an arbitrary position $1 \leq j_1 \leq 2m + 1$ to place $\mathcal{K}_1$, the first element of $\mathcal{K}$ (i.e., fix $p(j_1) = \mathcal{K}_1$), mark m additional positions among the first $2m + 1$ of p as *small*, and the remaining m as *large*. For $j > 2m + 1$, mark the j-th entry of the permutation as *small* if the vertex labelled j is positioned to the left of the topmost branching in H, and as *large* otherwise.

Step 3.2. Define new undetermined bijections $p^\pm$, where p^+ contains all the large positions of p, and p^- all the small ones. These bijections will have the ranges $\mathcal{R}^+ := \mathcal{R} \cap \{\mathcal{K}_1 + 1, \ldots, n\}$ and $\mathcal{R}^- := \mathcal{R} \cap \{1, \ldots, \mathcal{K}_1 - 1\}$, respectively. Moreover, let $\mathcal{K}^\pm$ be the subsequences of $\mathcal{K}$ containing, in the same order, the entries strictly larger/smaller than $\mathcal{K}_1$. Also split $\mathcal{H}$ into $\mathcal{H}^\pm$ such that $\mathcal{H}^-$ contains the vertices labelling the small positions in π and such that $\mathcal{H}^-$ below the m-th vertex is equal to the left subtree of $\mathcal{H}$ from Step 3.1. Construct $\mathcal{H}^+$ analogously, then relabel $\mathcal{H}^\pm$ with integers from $1, \ldots, |\mathcal{H}^\pm|$ while maintaining the relative order.

Step 3.3: Repeat steps 3.1–3.3 for both $(p^\pm, \mathcal{R}^\pm, \mathcal{K}^\pm, \mathcal{H}^\pm)$.

▶ **Lemma 8.** *If H comes from a topological ordering of $G(T, \pi^{(1)})$ and π_ι is constructed from T and $\pi^{(1)}$ as in step 1, then the following holds for step 3.2:*

(i) *$\mathcal{K}^\pm$ consists of those entries of π_ι that are contained in $\mathcal{R}^\pm$.*
(ii) *$|\mathcal{H}^\pm| = |\mathcal{R}^\pm| \geq m$, and each of $\mathcal{R}^\pm$ is a set of consecutive integers.*
(iii) *The lengths of $\mathcal{K}^\pm$ are equal to the number of branchings in $\mathcal{H}^\pm$.*

Moreover, the set of permutations constructible with step 3 is $\underline{\pi}(H)$.

▶ Remark 9. Since we have $\underline{\pi}(T) = \bigcup_H \underline{\pi}(H)$, where the union is disjoint and to be taken over all histories leading to T, this means we can construct $\underline{\pi}(T)$ out of $\underline{\pi}(T^{(1)})$ by performing steps 1–3 for all $\pi^{(1)}$ in $\underline{\pi}(T^{(1)})$.

3.3 An example

We give an example to illustrate the procedure: Suppose $m = 1$, $n = 9$, and consider the permutation $\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$. This permutation produces a *B*-tree T of the form given in Figure 2a – in fact, this permutation gives the history shown in Figure 1. Thus $T^{(1)}$ contains 4 keys, and $\pi^{(1)} = (1, 3, 4, 2) \in \underline{\pi}(T^{(1)})$.

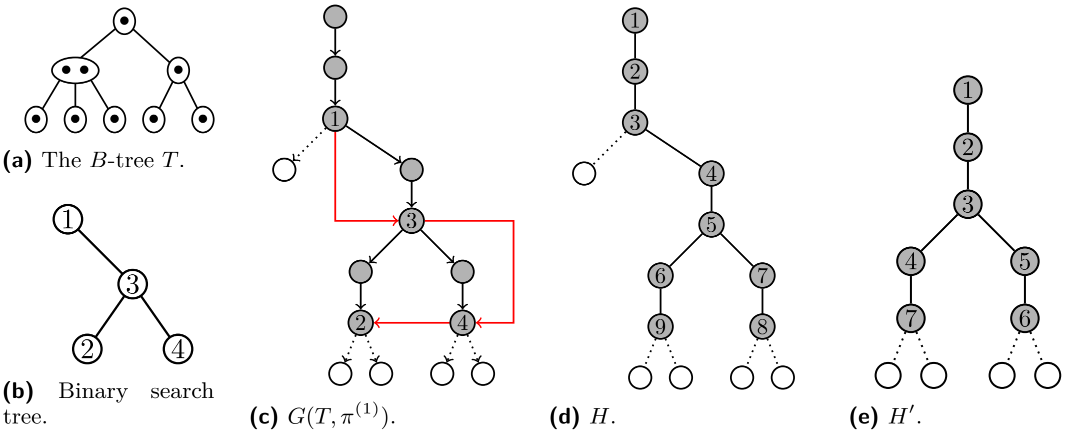
Step 1. The in-order traversal of T reveals that the keys in $T^{(1)}$ correspond to the keys $2, 4, 6, 8$ in T. Then the injection on the keys is given by $\{1, 2, 3, 4\} \ni i \mapsto 2i \in \{1, \dots, 9\}$, and applying this to the entries of π gives $\pi_\iota = (2, 6, 8, 4)$.

Step 2. Constructing a binary search tree from $\pi^{(1)}$ gives the one shown in Figure 2b which is then turned into the DAG $G = G(T, \pi^{(1)})$ shown in Figure 2c (the remaining labels are there to indicate how it connects to the binary search tree and to $\pi^{(1)}$). This graph has three distinct topological labellings, one of which induces the H depicted in Figure 2d.

Step 3. We initialize $p = (\_,\_,\_,\_,\_,\_,\_,\_,\_)$, $\mathcal{R} = \{1, \dots, 9\}$, $\mathcal{K} = (2, 6, 8, 4)$ and $\mathcal{H} = H$. After step 3.1, we have e.g. $p = (\ell, s, 2, \ell, \ell, \ell, \ell, \ell, \ell)$, where we write s for a small position, and ℓ for a large. Here, the assignment of $\ell, s, 2$ to the first 3 positions can be done arbitrarily (but we choose the options that will reconstruct π from above), the remainder is given by comparing it to H: the vertices labelled $4, 5, \dots$ are all positioned to the right of the top-most branching in H. This leads to $p^- = (\_)$ with $\mathcal{R}^- = \{1\}$, which in the next round of the recursion simply becomes $p = (1)$, and to $p^+ = (\_,\_,\_,\_,\_,\_,\_)$ with $\mathcal{R}^+ = \{3, \dots, 9\}$, $\mathcal{K}^+ = \{6, 8, 4\}$, and an $\mathcal{H}^+ = H'$ given by Figure 2e.

In the second round of the recursion (using the "+"-branch, as the other one is trivial), we have e.g. $p = (6, s, \ell, s, \ell, \ell, s)$, where the assignment of $\ell, s, 6$ to the first 3 positions can again be done arbitrarily, and the rest is governed by H'. This gives $p^\pm = (\_,\_,\_)$ with $\mathcal{R}^- = \{3, 4, 5\}, \mathcal{K}^- = (4)$ and $\mathcal{R}^+ = \{7, 8, 9\}, \mathcal{K}^+ = (8)$, respectively. Due to the small number of entries, both $\mathcal{H}^\pm$ are given by the unique 3-historic tree on 3 vertices. In the next two rounds of the recursion, the $p^\pm$ will then be filled by arbitrary assignments of the numbers in their range, say $p^+ = (7, 9, 8)$ and $p^- = (4, 5, 3)$.

Finally, we can put everything back together by embedding a pair of $p^\pm$ into the previous p according to the assignment of s and ℓ. Thus, $p^+ = (7, 9, 8)$ and $p^- = (4, 5, 3)$ together yield $(6, 4, 7, 5, 9, 8, 3)$. This in turn was p^+ from the first iteration of step 3, and together with the corresponding $p^- = (1)$, we regain $\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$.



(a) The B-tree T.

(b) Binary search tree.

(c) $G(T, \pi^{(1)})$.

(d) H.

(e) H'.

Figure 2 Steps in the algorithm of Section 3, as performed in Subsection 3.3.

3.4 Proofs

Proof of Lemma 7. For G to contain a directed cycle, we need two vertices $v_i = \pi^{(1)}(i)$ and $v_j = \pi^{(1)}(j)$ such that v_i is a descendant of v_j in the tree (i.e., according to the black edges), but v_j is a descendant of v_i according to the red edges. However, the latter only means that $i < j$. Accordingly, $\pi^{(1)}(i)$ was the first to be used for the binary search tree's construction, and hence v_i cannot be below v_j in the tree. Thus G is acyclic.

Trivially, any topological ordering of G yields an increasing labelling for the tree, so the induced H is historic. By construction of H, the final tree in the corresponding history $(T_1, \dots, T_n)$ will have the same leaves as T (according to Proposition 2(iii)), and thus the same set of keys in $T_n^{(1)}$ as in $T^{(1)}$. Moreover, by Lemma 6, these keys are moved upwards from the leaves in the relative order described by $\pi^{(1)}$, hence $T_n^{(1)} = T^{(1)}$ and H is a history of T that belongs to $\hat{\Psi}^{-1}(\pi^{(1)})$.

Conversely, consider now a history $(T_1, \dots, T_n) \in \hat{\Psi}^{-1}(\pi^{(1)})$ with the associated historic tree H_n. As in step 1, we obtain from $\pi^{(1)}$ and T_n a sequence $\pi_\iota = (K_{i_1}, \dots, K_{i_{n_1}})$, where the i_j are precisely the times in the history when a leaf was split. The historic tree H_n therefore must have its branchings labelled by the i_j (and this forces the labelling to be increasing along the red edges in G), and to be consistent with Lemma 6, the left/right-positioning of the branchings has to correspond to the one in a binary search tree obtained from π_ι or equivalently $\pi^{(1)}$. Thus, such an H_n is of the type constructed in Step 2, and its labelling is a topological labelling of G. ◀

Proof of Lemma 8. Claim (i) is evident from the construction. For (ii), the equality $|\mathcal{H}^\pm| = |\mathcal{R}^\pm|$ follows from Lemma 6, we have $|\mathcal{R}^\pm| \geq m$ since $|\mathcal{R}| \geq 2m + 1$, and consecutivity is again immediate from the construction. Claim (iii) again follows from Lemma 6. Taken together, these three claims ensure that the recursion in step 3 is well-defined whenever we initialize as in step 3.0.

For the final assertion, we first define $\pi^\pm$ to be permutations obtained from $p^\pm$ in step 3.2 by mapping $\mathcal{R}^\pm$ to $\{1, \dots, |\mathcal{R}^\pm|\}$ in an order-preserving fashion. We now use strong induction on n, where $T, \pi^{(1)}$, and H are arbitrary but coherent in the sense of Lemma 7. For $n \leq 2m$, there is nothing to show, as step 3.1 gives $\underline{\pi}(H) = S_n$.

For all larger n, observe that if $\pi \in \underline{\pi}(H)$ then K_{i_1} is the median of $\pi(1), \dots, \pi(2m+1)$ and $\pi^\pm \in \underline{\pi}(\mathcal{H}^\pm)$. The first property is equivalent to K_{i_1} being moved upwards from the leaves at the first branching of H and is ensured by step 3.1. The second property comes from

Lemma 6: The large entries in π are precisely (except for the first m) those corresponding to the right descendants of the first branching in H. This holds by step 3.2, and $\pi^{\pm}$ are constructible by the induction hypothesis since $|\mathcal{R}^{\pm}| < n$. Thus, such π is constructible.

Conversely, suppose π is constructible. Then, by the recursion, $\pi^{\pm}$ are constructible, and thus $\pi^{\pm} \in \underline{\pi}(\mathcal{H}^{\pm})$ by the induction hypothesis. The i-th entry of $\pi^{\pm}$ is simultaneously the i-th large/small entry (according to step 3.1) of π. For $i < m$, this is chosen arbitrarily among all possible configurations for the first $2m+1$ entries of π. For $i > m$, the entry in π is dictated by H, but is not among the first $2m+1$. Thus, it corresponds to a descendant of the first branching, and it follows from Lemma 6 that filling the large/small entries with the entries from $\pi^{\pm}$ in order produces a $\pi \in \underline{\pi}(H)$. ◀

Proof of Proposition 3. This lemma follows from an analysis of step 3. Indeed, whenever we are placing a $\mathcal{K}_1$ into the permutation in step 3.1, we have $2m+1$ choices for the exact position, and then $\binom{2m}{m}$ choices for the location of the small positions within the first $2m+1$ slots of p. Placing such a $\mathcal{K}_1$ corresponds exactly to the branchings in H, and it is clear that every possible choice will lead to a different π in the end. Whenever we have $|\mathcal{R}| \leq 2m$, we have $|\mathcal{R}|!$ choices. Moreover, invoking Lemma 8, the corresponding $\mathcal{K}$ is the empty sequence, hence the entries of $\mathcal{R}$ are the keys that end up in a joint leaf – say, the i-th leaf – of T. Thus by Proposition 2(iii), $|\mathcal{R}| = m + s_i$, and (1) follows. ◀

4 The number of histories

In this section, we will be interested in the number of possible histories that can arise, and in particular the asymptotic behaviour of this number. We focus on the case $m = 1$. In this case, a historic tree is a binary increasing tree where only vertices at even heights can have two children. Vertex 2 is always the only child of vertex 1, and vertex 3 is always the only child of vertex 2. It will be advantageous later to remove vertex 1 (and decrease all other labels by 1); we call the result a *reduced historic tree.* In such a tree, only vertices at odd heights can have two children.

If we remove the top two vertices (vertices 1 and 2, referred to in the following as the *stem*) from a reduced historic tree with n vertices, then it decomposes into two smaller reduced historic trees, each possibly only consisting of a single external vertex. On the level of generating functions, this translates to a second-order differential equation for the exponential generating function $H(x) = \sum_{n\geq 0} \frac{h_n}{n!} x^n$, where h_n is the number of reduced historic trees with n internal vertices (equivalently, the number of histories of length $n+1$). We have

$$H''(x) = H(x)^2, \qquad H(0) = H'(0) = 1. \tag{2}$$

One can compare this to the well-known differential equation $T'(x) = T(x)^2$ for the exponential generating function associated with arbitrary binary increasing trees, see for example [8, Lemma 6.4]. We remark here that the tree consisting only of a single external vertex is often not counted, in which case the equation becomes $T'(x) = (1 + T(x))^2$ instead. The sequence h_n (see [12, A007558]) and the associated differential equation (2) were analysed in a different context in [5]: the differential equation has an explicit solution that can be expressed in terms of the Weierstrass elliptic function. It has a dominant singularity at $\rho \approx 2.3758705509$ where

$$H(x) \sim \frac{c}{(1 - x/\rho)^2}$$

for a constant $c = 6\rho^{-2} \approx 1.0629325375$. This leads to the following asymptotic behaviour:

$$\frac{h_n}{n!} \sim cn\rho^{-n} = 6n\rho^{-n-2}.$$

We would now like to generalise the differential equation to arbitrary $m \geq 1$. We remove the first m vertices from a $(2m+1)$-historic tree to obtain a reduced $(2m+1)$-historic tree, which is now a binary increasing tree where only vertices at heights $\equiv -1 \mod m+1$ can have two children. Removing the stem consisting of $m+1$ vertices decomposes such a tree into two smaller trees with the same property (each of them can also be a single external vertex). So in analogy to (2), we obtain a differential equation of order $m+1$, namely

$$H^{(m+1)}(x) = H(x)^2, \qquad H(0) = H'(0) = \cdots = H^{(m)}(0) = 1. \tag{3}$$

This higher-order differential equation can no longer be solved in an explicit fashion, as it was the case for $m = 1$. If we assume that there is a dominant singularity ρ_m where the behaviour of H is of the form $c_m(1 - x/\rho_m)^{-a_m}$, then comparing the two sides of the equation gives us

$$c_m \frac{a_m(a_m+1)\cdots(a_m+m)}{\rho_m^{m+1}}(1 - x/\rho_m)^{-a_m-m-1} = c_m^2(1 - x/\rho_m)^{-2a_m},$$

thus $a_m = m+1$ and $c_m = \frac{(2m+1)!}{m!}\rho_m^{-m-1}$. Applying singularity analysis would then yield

$$[x^n]H(x) \sim \frac{c_m}{m!}n^m\rho_m^{-n} = \frac{(2m+1)!}{(m!)^2}n^m\rho_m^{-n-m-1}.$$

This leads us to the following conjecture:

▶ **Conjecture 10.** *For every $m \geq 1$, the number of reduced $(2m+1)$-historic trees with n vertices (corresponding to histories of length $n+m$) is asymptotically equal to*

$$n! \cdot \frac{(2m+1)!}{(m!)^2}n^m\rho_m^{-n-m-1}$$

for some positive constant ρ_m.

Numerical evidence for small values of m seems to support this conjecture, as the fit of the asymptotic formula with the actual coefficients is excellent. Experimental values of the exponential growth rate ρ_m^{-1} are given in Table 1.

Table 1 Experimental values of ρ_m^{-1} for $2 \leq m \leq 6$.

m	2	3	4	5	6
ρ_m^{-1}	3.7746	5.1792	6.5857	7.9928	9.3999

5 Statistics of *B*-trees via historic trees

Let us now study B-trees that are constructed by successive insertion of n random numbers. Equivalently, we can think of them as being constructed from a random permutation of $1, 2, \ldots, n$. In order to apply the connection to historic trees, we need to take the number of permutations associated with a specific history into account.

Again, we focus on the special case $m = 1$. Proposition 3 tells us that the number of permutations corresponding to a specific historic tree T is in this case $6^{\mathsf{b}(T)}2^{\mathsf{i}(T)}$, where $\mathsf{b}(T)$ is the number of branchings and $\mathsf{i}(T)$ the number of internal vertices that lie directly between a branching and an external vertex. This remains true if we consider reduced historic trees. We associate this number as a weight $w(T) = 6^{\mathsf{b}(T)} 2^{\mathsf{i}(T)}$ with every reduced historic tree T and consider the weighted exponential generating function (rather than the unweighted one that was analysed in the previous section). In the recursive decomposition of a reduced historic tree into its stem and two smaller trees T_1 and T_2, we have $\mathsf{b}(T) = \mathsf{b}(T_1) + \mathsf{b}(T_2) + 1$ and $\mathsf{i}(T) = \mathsf{i}(T_1) + \mathsf{i}(T_2)$. This is even true if T_1 or T_2 (or both) only consist of a single external vertex. Thus we obtain

$$w(T) = 6^{\mathsf{b}(T_1)+\mathsf{b}(T_2)+1} \cdot 2^{\mathsf{i}(T_1)+\mathsf{i}(T_2)} = 6w(T_1)w(T_2). \tag{4}$$

On the level of the weighted exponential generating function $W(x)$, (2) becomes

$$W''(x) = 6W(x)^2, \qquad W(0) = 1, W'(0) = 2. \tag{5}$$

Unlike (2), however, there is now a very simple explicit solution, namely $W(x) = \frac{1}{(1-x)^2}$. This is not unexpected, since the total weight of all reduced historic trees with n internal vertices must be equal to the number of permutations of $1, 2, \dots, n+1$. Thus

$$W(x) = \sum_{n \geq 0} \frac{(n+1)!}{n!} x^n = \frac{1}{(1-x)^2}.$$

Recall that the number of external vertices in n-vertex reduced 3-historic trees is in bijection with the number of leaves in 2-3-trees built from $n+1$ keys. Thus, as a next step, we incorporate the number of external vertices $\mathsf{e}(T)$ as an additional statistic in our generating function in order to prove the following theorem:

▶ **Theorem 11.** *Let L_n be the number of leaves in a 2-3-tree built from n random keys. Then we have $\mathbb{E}(L_n) = \frac{3}{7}(n+1)$ and $\mathbb{V}(L_n) = \frac{12}{637}(n+1)$ for $n > 11$. Moreover, the central limit theorem*

$$\frac{L_n - \mathbb{E}(L_n)}{\sqrt{\mathbb{V}(L_n)}} \xrightarrow{\mathrm{d}} N(0,1)$$

holds.

Proof. Let us consider the bivariate generating function in which the second variable u marks the number of external vertices $\mathsf{e}(T)$:

$$W(x,u) = \sum_{T} \frac{1}{|T|!} x^{|T|} u^{\mathsf{e}(T)}.$$

Since $\mathsf{e}(T) = \mathsf{e}(T_1) + \mathsf{e}(T_2)$, the differential equation (5) is actually unaffected by the additional variable; the only change concerns the initial values. We have (where derivatives are taken with respect to x)

$$W''(x,u) = 6W(x,u)^2, \qquad W(0,u) = u, W'(0,u) = 2u, \tag{6}$$

which no longer has an equally simple explicit solution. Using the method described in [5], it can, however, be expressed as the inverse function to

$$X(w,u) = \int_u^w \frac{1}{\sqrt{4t^3 + 4u^2(1-u)}}\, dt.$$

It follows that $W(x,u)$ has a dominant singularity at $\rho(u) = \int_u^\infty \frac{1}{\sqrt{4t^3+4u^2(1-u)}}\,dt$: as $w \to \infty$, we have

$$X(w,u) = \rho(u) - \frac{1}{\sqrt{w}} + O(w^{-7/2}),$$

thus

$$W(x,u) \sim \frac{1}{(\rho(u)-x)^2}$$

at the singularity. An application of the quasi-power theorem [9, Theorem IX.8] yields a central limit theorem for the number of external vertices. Moreover, one can obtain explicit expressions for the moments. Differentiating (6) with respect to u and plugging in $u=1$, we obtain the following differential equation for $W_1(x) = \frac{\partial}{\partial u}W(x,u)\Big|_{u=1}$:

$$W_1''(x) = 12W(x,1)W_1(x) = \frac{12}{(1-x)^2}W_1(x), \qquad W_1(0) = 1, W_1'(0) = 2,$$

since we already know that $W(x,1) = W(x) = (1-x)^{-2}$. This linear differential equation has the two linearly independent solutions $(1-x)^{-3}$ and $(1-x)^4$, and one obtains

$$W_1(x) = \frac{6}{7(1-x)^3} + \frac{(1-x)^4}{7}.$$

Thus for $n > 4$, we have $[x^n]W_1(x) = \frac{6}{7}[x^n](1-x)^{-3} = \frac{6}{7}\binom{n+2}{2}$. Consequently, the average number of external vertices is

$$\frac{[x^n]W_1(x)}{[x^n]W(x)} = \frac{\frac{6}{7}\binom{n+2}{2}}{n+1} = \frac{3(n+2)}{7}.$$

In the same way, one can treat the second moment: to this end, we consider $W_2(x) = \left(\frac{\partial}{\partial u}\right)^2 W(x,u)\Big|_{u=1}$. Differentiating (6) twice with respect to u and plugging in $u=1$ now gives us

$$\begin{aligned} W_2''(x) &= 12W(x)W_2(x) + 12W_1(x)^2 \\ &= \frac{12}{(1-x)^2}W_2(x) + 12\Big(\frac{6}{7(1-x)^3} + \frac{(1-x)^4}{7}\Big)^2, \qquad W_2(0) = W_2'(0) = 0. \end{aligned}$$

The solution to this differential equation is given by

$$W_2(x) = \frac{54}{49(1-x)^4} - \frac{108}{91(1-x)^3} - \frac{24}{49}(1-x)^3 + \frac{4}{7}(1-x)^4 + \frac{2}{637}(1-x)^{10}.$$

So for $n > 10$, $[x^n]W_2(x) = \frac{54}{49}\binom{n+3}{3} - \frac{108}{91}\binom{n+2}{2} = \frac{9(n+1)(n+2)(13n-3)}{637}$. It follows that the variance of the number of external vertices is

$$\frac{[x^n](W_1(x)+W_2(x))}{[x^n]W(x)} - \Big(\frac{[x^n]W_1(x)}{[x^n]W(x)}\Big)^2 = \frac{12(n+2)}{637}.$$

This completes the proof. ◀

The approach in our proof provides an alternative to the analysis via Pólya urns, see [1–3, 13] (in particular, the mean was first determined by Yao [13] by explicitly solving a recursion). Here, one can think of the leaves in a B-tree as balls in an urn of different types depending on the number of keys they hold. Adding a new key then corresponds to picking a ball from the urn and replacing it by a new ball (of different type), or two new balls in the case of a node split.

The same calculations for the moments as in Theorem 11 can also be carried out for higher values of m, though the expressions become more complicated. For general $m \geq 1$, the differential equation becomes

$$W^{(m+1)}(x,u) = \frac{(2m+1)!}{m!^2} W(x,u)^2,$$

with initial values

$$W^{(i)}(0,u) = (m+i)!u, \qquad i = 0,1,\dots,m.$$

In particular, we have $W(x) = W(x,1) = m!(1-x)^{-m-1}$, and $W_1(x) = \frac{\partial}{\partial u} W(x,u)\Big|_{u=1}$ satisfies the linear differential equation

$$W_1^{(m+1)}(x) = \frac{2(2m+1)!}{m!}(1-x)^{-m-1} W_1(x).$$

Up to a trivial change of variables (substituting for $1-x$), this is a linear differential equation of Cauchy–Euler type that can be solved with standard tools. In fact, setting $e^{-t} = 1-x$ turns it into a linear differential equation with constant coefficients. Functions of the form $f(x) = (1-x)^{-b}$ with

$$b^{\overline{m+1}} = b(b+1)\cdots(b+m) = \frac{2(2m+1)!}{m!} = \frac{(2m+2)!}{(m+1)!} \tag{7}$$

are particular solutions to this differential equation. Note that $b = m+2$ is always a solution to (7). The general solution can be determined as linear combination of particular solutions, taking the initial values into account. The term of the form $c(1-x)^{-m-2}$ in $W_1(x)$ dominates asymptotically.

To give one more concrete example, for $m = 2$ we have the differential equation

$$W'''(x,u) = 30W(x,u)^2, \qquad W(0,u) = 2u, W'(0,u) = 6u, W''(0,u) = 24u.$$

Thus $W(x,1) = 2(1-x)^{-3}$. The solutions to (7) are now $b = 4$ and $b = \frac{-7\pm\sqrt{71}i}{2}$. Taking the initial values into account, we obtain

$$W_1(x) = \frac{60}{37(1-x)^4} + \frac{7\sqrt{71}+31i}{37\sqrt{71}}(1-x)^{(7+\sqrt{71}i)/2} + \frac{7\sqrt{71}-31i}{37\sqrt{71}}(1-x)^{(7-\sqrt{71}i)/2}.$$

Thus the average number of external vertices in reduced 5-historic trees with n vertices is $\frac{10(n+3)}{37} + O(n^{-13/2})$.

For general m, one finds from the differential equation that the function $\ell(t) = W_1(1-e^{-t})$ has Laplace transform

$$L(s) = \frac{m!((m+1)^{\overline{m+1}} - s^{\overline{m+1}})}{(s-m-1)((m+2)^{\overline{m+1}} - s^{\overline{m+1}})},$$

where $s^{\overline{h}} = s(s+1)\cdots(s+h-1)$ is a rising factorial as in (7). The term $\frac{\kappa_m}{s-m-2}$ in the partial fraction decomposition corresponds to the dominant term

$$W_1(x) \sim \frac{\kappa_m}{(1-x)^{m+2}}.$$

Here we have, with $H_k = 1 + \frac{1}{2} + \cdots + \frac{1}{k}$ denoting a harmonic number,

$$\kappa_m = \frac{m!}{2(H_{2m+2} - H_{m+1})}.$$

Consequently, the average number of external vertices in reduced $(2m+1)$-historic trees with n vertices is asymptotically equal to $\frac{\kappa_m}{(m+1)!} \cdot n = \frac{1}{2(m+1)(H_{2m+2}-H_{m+1})} \cdot n$. Some explicit values of the constant $\frac{\kappa_m}{(m+1)!}$ are given in Table 2.

Table 2 Values of the constant $\frac{\kappa_m}{(m+1)!} = \frac{1}{2(m+1)(H_{2m+2}-H_{m+1})}$ for $1 \leq m \leq 10$.

m	1	2	3	4	5	6	7	8	9	10
$\frac{\kappa_m}{(m+1)!}$	$\frac{3}{7}$	$\frac{10}{37}$	$\frac{105}{533}$	$\frac{252}{1627}$	$\frac{2310}{18107}$	$\frac{25740}{237371}$	$\frac{9009}{95549}$	$\frac{136136}{1632341}$	$\frac{11639628}{155685007}$	$\frac{10581480}{156188887}$

6 Conclusion and perspective

The connection between B-tree histories and historic trees provides us with a novel way to analyse B-trees and their evolution. Possible future directions include studying further statistics of B-trees and historic trees and considering higher values of m. In particular, a proof of Conjecture 10 would be desirable. It might even be interesting, at least from a purely mathematical perspective, to allow m to grow with n.

References

1 D. Aldous, B. Flannery, and J.L. Palacios. Two applications of urn processes. *Probability in the Engineering and Informational Sciences*, 2(3):293–307, 1988.

2 Ricardo A. Baeza-Yates. Fringe analysis revisited. *ACM Comput. Surv.*, 27(1):111–119, 1995. doi:10.1145/214037.214103.

3 A. Bagchi and A.K. Pal. Asymptotic normality in the generalized Pólya–Eggenberger urn model, with an application to computer data structures. *SIAM Journal on Algebraic Discrete Methods*, 6(3):394–405, 1985.

4 R. Bayer and E. McCreight. Organization and maintenance of large ordered indexes. *Acta Informatica*, 1:173–189, 1972.

5 Olivier Bodini, Matthieu Dien, Xavier Fontaine, Antoine Genitrini, and Hsien-Kuei Hwang. Increasing diamonds. In *LATIN 2016: theoretical informatics*, volume 9644 of *Lecture Notes in Comput. Sci.*, pages 207–219. Springer, Berlin, 2016. doi:10.1007/978-3-662-49529-2_16.

6 B. Chauvin, D. Gardy, N. Pouyanne, and D.-H. Ton-That. B-urns. *ALEA. Latin American Journal of Probability & Mathematical Statistics*, 13:605–634, 2016.

7 V. Chvátal, D.A. Klarner, and D.E. Knuth. Selected combinatorial research problems. *Computer Science Department, Stanford University*, STAN-CS-72-292, 1972.

8 Michael Drmota. *Random trees.* SpringerWienNewYork, Vienna, 2009. doi:10.1007/978-3-211-75357-6.

9 Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics.* Cambridge University Press, 2009.

10 D.E. Knuth. *The Art of Computer Programming: Volume 3. Sorting and Searching.* Addison-Wesley, 2nd edition, 1998.

11 A.M. Odlyzko. Periodic oscillations of coefficients of power series that satisfy functional equations. *Advances in Mathematics*, 44:180–205, 1982.

12 OEIS Foundation Inc. The On-Line Encyclopedia of Integer Sequences, 2024. Published electronically at https://oeis.org.

13 A.C.C. Yao. On random 2–3 trees. *Acta informatica*, 9(2):159–170, 1978.