Distinct fringe subtrees in binary search trees

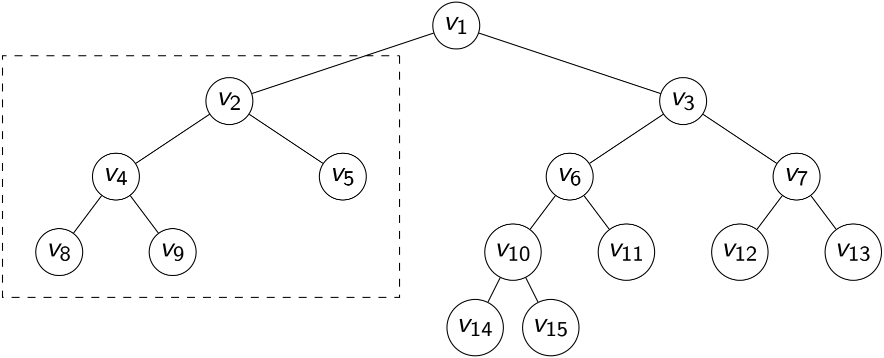
Stephan Wagner

TU Graz and Uppsala University

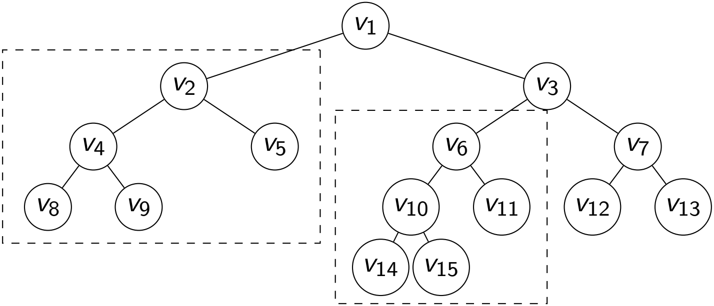
AofA2024 Bath, 18 June 2024

Fringe subtrees

A *fringe subtree* of a rooted tree is a subtree that consists of a vertex and all its descendants.

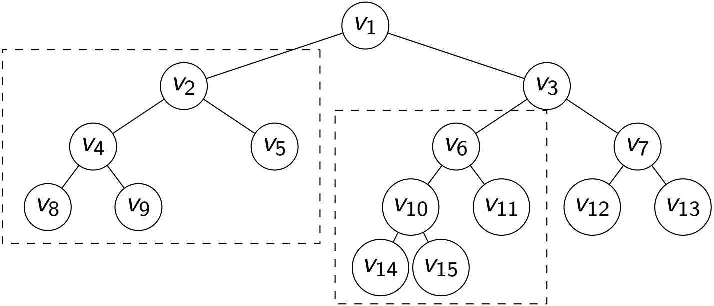


Identical and distinct fringe subtrees



The fringe subtrees rooted at v_2 and v_6 are identical as unlabelled plane trees.

Identical and distinct fringe subtrees



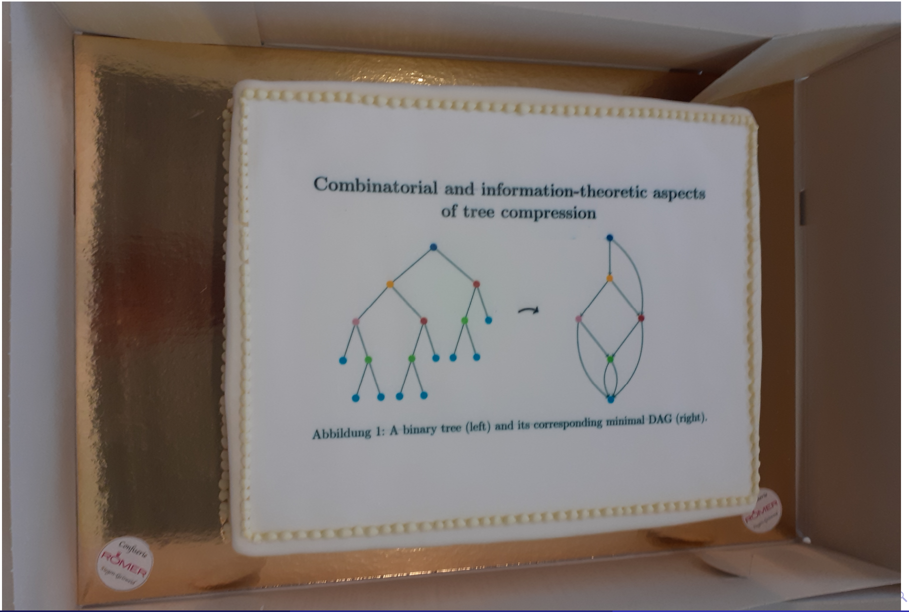
The fringe subtrees rooted at v_2 and v_6 are identical as unlabelled plane trees.

There are five distinct equivalence classes of fringe subtrees:

v_1 $\quad$ v_3 $\quad$ v_2, v_6 $\quad$ v_4, v_7, v_{10} $\quad$ $v_5, v_8, v_9, v_{11}, v_{12}, v_{13}, v_{14}, v_{15}$

Tree compression

Tree compression



Tree compression

Tree compression plays a role in (among other things)

- XML compression and querying,
- symbolic model checking,
- compiler construction.

Tree compression

Tree compression plays a role in (among other things)

- XML compression and querying,
- symbolic model checking,
- compiler construction.

The number of distinct fringe subtrees is a measure for how much a tree is compressed by constructing the minimal DAG.

Prior results: simply generated trees

Theorem (Flajolet/Sipala/Steyaert 1990; Seelbach Benkner/W 2022)

Let X_n be the number of distinct fringe subtrees in a random tree with n vertices from a simply generated family (with some technical conditions). Then we have

$$\mathbb{E}(X_n) \sim \frac{Cn}{\sqrt{\log n}}$$

for some constant C. Moreover,

$$\frac{X_n}{n/\sqrt{\log n}} \xrightarrow{p} C.$$

Prior results: simply generated trees

Theorem (Flajolet/Sipala/Steyaert 1990; Seelbach Benkner/W 2022)

Let X_n be the number of distinct fringe subtrees in a random tree with n vertices from a simply generated family (with some technical conditions). Then we have

$$\mathbb{E}(X_n) \sim \frac{Cn}{\sqrt{\log n}}$$

for some constant C. Moreover,

$$\frac{X_n}{n/\sqrt{\log n}} \xrightarrow{p} C.$$

For example, in the special case of uniformly random binary trees with n leaves ($n-1$ internal vertices), we have $\mathbb{E}(X_n) \sim \frac{2n}{\sqrt{\pi \log_4 n}}$.

In a binary search tree, the labels of the internal vertices are such that

Binary search trees

In a binary search tree, the labels of the internal vertices are such that

- all numbers less than the root label are in the left branch,

Binary search trees

In a binary search tree, the labels of the internal vertices are such that

- all numbers less than the root label are in the left branch,
- while all numbers greater than the root label are in the right branch.

Binary search trees

In a binary search tree, the labels of the internal vertices are such that

- all numbers less than the root label are in the left branch,
- while all numbers greater than the root label are in the right branch.

We will be interested in *random binary search trees* built from a random permutation of $\{1, 2, \ldots, n\}$.

Binary search trees

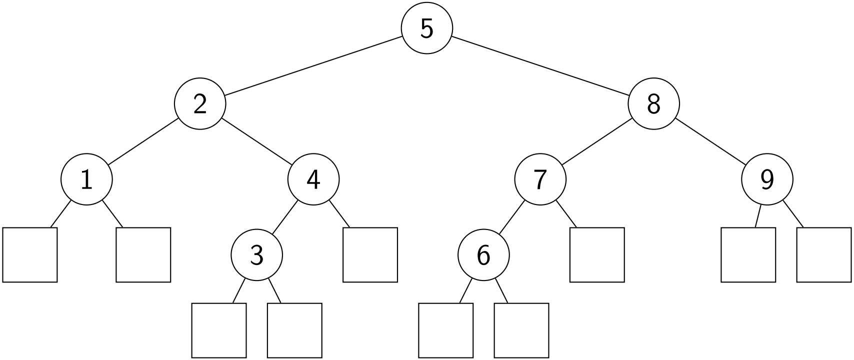
In a binary search tree, the labels of the internal vertices are such that

- all numbers less than the root label are in the left branch,
- while all numbers greater than the root label are in the right branch.

We will be interested in *random binary search trees* built from a random permutation of $\{1, 2, \ldots, n\}$.

It is well known that this random tree model is also essentially equivalent to that of *binary increasing trees*, where vertex labels are increasing from the root to the leaves.

Binary search trees



Binary search tree resulting from the permutation $(5, 2, 8, 4, 1, 7, 9, 3, 6)$.

Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

Prior results: binary search trees

Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

- Flajolet/Gourdon/Martínez 1997: $\mathbb{E}(F_n) \leq \frac{(4\log 2)n}{\log n}(1 + o(1))$

Prior results: binary search trees

Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

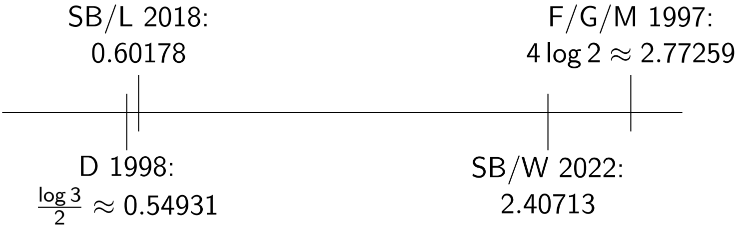
- Flajolet/Gourdon/Martínez 1997: $\mathbb{E}(F_n) \leq \frac{(4 \log 2)n}{\log n}(1 + o(1))$
- Devroye 1998: $\mathbb{E}(F_n) \geq \frac{(\log 3)n}{2 \log n}(1 + o(1))$

Prior results: binary search trees

Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

- Flajolet/Gourdon/Martínez 1997: $\mathbb{E}(F_n) \leq \frac{(4 \log 2)n}{\log n}(1 + o(1))$
- Devroye 1998: $\mathbb{E}(F_n) \geq \frac{(\log 3)n}{2 \log n}(1 + o(1))$
- Seelbach Benkner/Lohrey 2018: $\mathbb{E}(F_n) \geq 0.60178 \frac{n}{\log n}(1 + o(1))$

Prior results: binary search trees



Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

- Flajolet/Gourdon/Martínez 1997: $\mathbb{E}(F_n) \leq \frac{(4\log 2)n}{\log n}(1+o(1))$
- Devroye 1998: $\mathbb{E}(F_n) \geq \frac{(\log 3)n}{2\log n}(1+o(1))$
- Seelbach Benkner/Lohrey 2018: $\mathbb{E}(F_n) \geq 0.60178\frac{n}{\log n}(1+o(1))$
- Seelbach Benkner/W 2022: $\mathbb{E}(F_n) \geq 2.40713\frac{n}{\log n}(1+o(1))$

Prior results: binary search trees

Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices.

- Flajolet/Gourdon/Martínez 1997: $\mathbb{E}(F_n) \leq \frac{(4 \log 2)n}{\log n}(1 + o(1))$
- Devroye 1998: $\mathbb{E}(F_n) \geq \frac{(\log 3)n}{2 \log n}(1 + o(1))$
- Seelbach Benkner/Lohrey 2018: $\mathbb{E}(F_n) \geq 0.60178\frac{n}{\log n}(1 + o(1))$
- Seelbach Benkner/W 2022: $\mathbb{E}(F_n) \geq 2.40713\frac{n}{\log n}(1 + o(1))$

Main theorem

Theorem

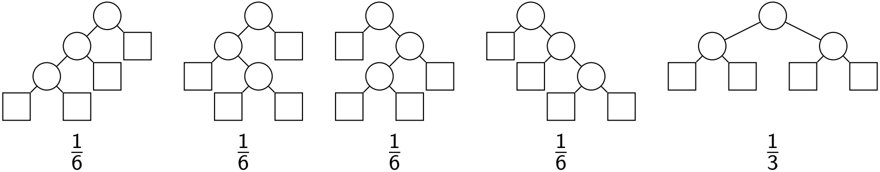
Let F_n be the number of distinct fringe subtrees in a random binary search tree with n internal vertices, and let c_1 be the constant $4\sum_{k\geq 1}\frac{\log k}{(k+1)(k+2)} \approx 2.40713$. We have

$$\mathbb{E}(F_n) \sim \frac{c_1 n}{\log n}$$

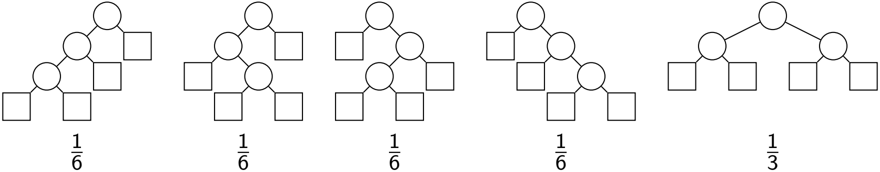
as $n \to \infty$. Moreover, we also have convergence in probability:

$$\frac{F_n}{n/\log n} \xrightarrow{p} c_1.$$

Binary search tree distribution



Binary search tree distribution



The probability that a random binary search tree has a specific shape T can be expressed as

$$p(T) = \prod_{v} \frac{1}{N_v},$$

where the product is over all internal vertices and N_v is the number of *internal vertices* in the fringe subtree rooted at v.

Shape functional

The negative logarithm of $p(T)$, which can be expressed as

$$-\log p(T) = \sum_v \log N_v,$$

is called the *shape functional* of T.

Shape functional

The negative logarithm of $p(T)$, which can be expressed as

$$-\log p(T) = \sum_v \log N_v,$$

is called the *shape functional* of T.

Theorem (Fill 1996)

Let the random variable L_n be defined by $L_n = -\log p(\mathcal{T}_n)$, where $\mathcal{T}_n$ is a random binary search tree of size n (n external vertices). We have

$$\mathbb{E}(L_n) = \mu n + O(\log n),$$

where $\mu = \sum_{k=1}^{\infty} \frac{2\log k}{(k+1)(k+2)}$. Moreover, $\mathbb{V}(L_n) = \sigma^2 n + O(1)$ for a constant $\sigma^2 > 0$, and the centred and normalised random variable $\frac{L_n - \mu n}{\sigma\sqrt{n}}$ converges in distribution to a standard normal distribution.

Fringe subtrees of binary search trees

Facts about fringe subtrees of random binary search trees:

Fringe subtrees of binary search trees

Facts about fringe subtrees of random binary search trees:

- The expected number of fringe subtrees of size k in a random binary search tree of size n is $\frac{2n}{k(k+1)}$ for all $k < n$.

Fringe subtrees of binary search trees

Facts about fringe subtrees of random binary search trees:

- The expected number of fringe subtrees of size k in a random binary search tree of size n is $\frac{2n}{k(k+1)}$ for all $k < n$.
- Conditioned on its size, every fringe subtree is again a random binary search tree.

Fringe subtrees of binary search trees

Facts about fringe subtrees of random binary search trees:

- The expected number of fringe subtrees of size k in a random binary search tree of size n is $\frac{2n}{k(k+1)}$ for all $k < n$.
- Conditioned on its size, every fringe subtree is again a random binary search tree.
- So if $p_k = \sum_{B \in \mathfrak{S}_k} p(B)$ is the probability that a random binary search tree has a shape that belongs to some subset $\mathfrak{S}_k$ of the set $\mathfrak{B}_k$ of all binary trees of size k, then the expected number of fringe subtrees whose shape belongs to $\mathfrak{S}_k$ is $\frac{2p_k n}{k(k+1)}$.

Fringe subtrees of binary search trees

Facts about fringe subtrees of random binary search trees:

- The expected number of fringe subtrees of size k in a random binary search tree of size n is $\frac{2n}{k(k+1)}$ for all $k < n$.
- Conditioned on its size, every fringe subtree is again a random binary search tree.
- So if $p_k = \sum_{B \in \mathfrak{S}_k} p(B)$ is the probability that a random binary search tree has a shape that belongs to some subset $\mathfrak{S}_k$ of the set $\mathfrak{B}_k$ of all binary trees of size k, then the expected number of fringe subtrees whose shape belongs to $\mathfrak{S}_k$ is $\frac{2p_k n}{k(k+1)}$.
- Moreover, the number in the previous statement is concentrated around its mean.

Proof sketch: lower bound

- Focus on "large" trees whose size at least $k_0 := \frac{1}{\mu}(\log n + (\log n)^{3/4})$, where $\mu = \sum_{k=1}^{\infty} \frac{2 \log k}{(k+1)(k+2)}$.

Proof sketch: lower bound

- Focus on "large" trees whose size at least $k_0 := \frac{1}{\mu}(\log n + (\log n)^{3/4})$, where $\mu = \sum_{k=1}^{\infty} \frac{2 \log k}{(k+1)(k+2)}$.
- For $k \geq k_0$, choose $\mathfrak{S}_k$ to be the subset of $\mathfrak{B}_k$ consisting of those trees B for which $p(B) \leq \exp(-\mu k + k^{2/3})$, or equivalently $-\log p(B) \geq \mu k - k^{2/3}$.

Proof sketch: lower bound

- Focus on "large" trees whose size at least $k_0 := \frac{1}{\mu}(\log n + (\log n)^{3/4})$, where $\mu = \sum_{k=1}^{\infty} \frac{2\log k}{(k+1)(k+2)}$.
- For $k \geq k_0$, choose $\mathfrak{S}_k$ to be the subset of $\mathfrak{B}_k$ consisting of those trees B for which $p(B) \leq \exp(-\mu k + k^{2/3})$, or equivalently $-\log p(B) \geq \mu k - k^{2/3}$.
- Observe that binary search trees of size k belong to $\mathfrak{S}_k$ with high probability: $p_k = 1 - O(k^{-1/3})$.

Proof sketch: lower bound

- Focus on "large" trees whose size at least $k_0 := \frac{1}{\mu}(\log n + (\log n)^{3/4})$, where $\mu = \sum_{k=1}^{\infty} \frac{2\log k}{(k+1)(k+2)}$.
- For $k \geq k_0$, choose $\mathfrak{S}_k$ to be the subset of $\mathfrak{B}_k$ consisting of those trees B for which $p(B) \leq \exp(-\mu k + k^{2/3})$, or equivalently $-\log p(B) \geq \mu k - k^{2/3}$.
- Observe that binary search trees of size k belong to $\mathfrak{S}_k$ with high probability: $p_k = 1 - O(k^{-1/3})$.
- Show that most trees belonging to $\mathfrak{S}_k$ for some $k \geq k_0$ only occur at most once as a fringe subtree with high probability.

Proof sketch: lower bound

- Focus on "large" trees whose size at least $k_0 := \frac{1}{\mu}(\log n + (\log n)^{3/4})$, where $\mu = \sum_{k=1}^{\infty} \frac{2\log k}{(k+1)(k+2)}$.
- For $k \geq k_0$, choose $\mathfrak{S}_k$ to be the subset of $\mathfrak{B}_k$ consisting of those trees B for which $p(B) \leq \exp(-\mu k + k^{2/3})$, or equivalently $-\log p(B) \geq \mu k - k^{2/3}$.
- Observe that binary search trees of size k belong to $\mathfrak{S}_k$ with high probability: $p_k = 1 - O(k^{-1/3})$.
- Show that most trees belonging to $\mathfrak{S}_k$ for some $k \geq k_0$ only occur at most once as a fringe subtree with high probability.
- So the number of fringe subtrees whose size is at least k_0 provides an asymptotic lower bound:

$$F_n \gtrsim \sum_{k \geq k_0} \frac{2n}{k(k+1)} \sim \frac{2n}{k_0} \sim \frac{2\mu n}{\log n}.$$

Proof sketch: upper bound

- Split into "small", "medium" and "large" fringe subtrees:
  - Small: $k \leq k_1 := \frac{1}{2}\log_4 n$;
  - Medium: $k_1 < k \leq k_2 := \frac{1}{\mu}(\log n - (\log n)^{3/4})$, with μ as before.
  - Large: $k_2 < k$.

Proof sketch: upper bound

- Split into "small", "medium" and "large" fringe subtrees:
  - Small: $k \leq k_1 := \frac{1}{2} \log_4 n$;
  - Medium: $k_1 < k \leq k_2 := \frac{1}{\mu}(\log n - (\log n)^{3/4})$, with μ as before.
  - Large: $k_2 < k$.
- Bound the contribution of small fringe subtrees by the total number of possible binary trees of size $\leq k_1$.

Proof sketch: upper bound

- Split into "small", "medium" and "large" fringe subtrees:
  - Small: $k \leq k_1 := \frac{1}{2} \log_4 n$;
  - Medium: $k_1 < k \leq k_2 := \frac{1}{\mu}(\log n - (\log n)^{3/4})$, with μ as before.
  - Large: $k_2 < k$.
- Bound the contribution of small fringe subtrees by the total number of possible binary trees of size $\leq k_1$.
- Show that medium-sized fringe subtrees can be divided further into two parts:
  - a majority of trees with "large" shape functional—their probability to occur is too low for them to contribute asymptotically;
  - and a minority of trees with "small" shape functional—there are not enough of those to contribute asymptotically.

Proof sketch: upper bound

- Split into "small", "medium" and "large" fringe subtrees:
  - Small: $k \leq k_1 := \frac{1}{2}\log_4 n$;
  - Medium: $k_1 < k \leq k_2 := \frac{1}{\mu}(\log n - (\log n)^{3/4})$, with μ as before.
  - Large: $k_2 < k$.
- Bound the contribution of small fringe subtrees by the total number of possible binary trees of size $\leq k_1$.
- Show that medium-sized fringe subtrees can be divided further into two parts:
  - a majority of trees with "large" shape functional—their probability to occur is too low for them to contribute asymptotically;
  - and a minority of trees with "small" shape functional—there are not enough of those to contribute asymptotically.
- Bound the contribution of large fringe subtrees by their total number (ignoring whether they are distinct or not).

Entropy

The expected value of the shape functional can also be thought of as the entropy of the shape of a random binary search tree $\mathcal{T}_n$:

$$\mathbb{E}(L_n) = \mathbb{E}(-\log p(\mathcal{T}_n)) = -\sum_{B \in \mathfrak{B}_n} p(B) \log p(B).$$

So the growth constant for the number of distinct fringe subtrees is directly connected to the growth constant for this entropy.

Generalisations

The method is fairly general and also works for other types of random trees and notions of distinctness, provided that we have two ingredients available:

- information on the distribution of the number of fringe subtrees of a given size,
- information on the distribution of a suitable analogue of the shape functional.

Thank you!