A bijection for the evolution of B-trees

Fabian Burghart
Eindhoven University of Technology

Joint work with

Stephan Wagner
TU Graz & Uppsala University

June 18, 2024

Bath, AofA2024

What are B-trees?

Search trees, i.e. the nodes store *keys*, sorted by their size.
Nice property: B-trees are balanced.

What are B-trees?

Search trees, i.e. the nodes store *keys*, sorted by their size.
Nice property: B-trees are balanced.
We consider B-trees of order $2m + 1$: a node cannot contain more than $2m$ keys. Usually (in this talk) $m = 1$.

What are B-trees?

Search trees, i.e. the nodes store *keys*, sorted by their size.
Nice property: B-trees are balanced.
We consider B-trees of order $2m + 1$: a node cannot contain more than $2m$ keys. Usually (in this talk) $m = 1$.
B-trees can be constructed from a list of keys by an *insertion algorithm*.

What are B-trees?

Search trees, i.e. the nodes store *keys*, sorted by their size.
Nice property: B-trees are balanced.
We consider B-trees of order $2m + 1$: a node cannot contain more than $2m$ keys. Usually (in this talk) $m = 1$.
B-trees can be constructed from a list of keys by an *insertion algorithm*.
There are two possible ways to generate keys:

- Sampled from a continuous probability distribution (e.g. uniform on $[0, 1]$)
- Key sequence is a uniform permutation $\pi \in S_n$

Insertion algorithm, $m = 1$

$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

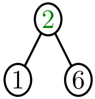
Insertion algorithm, $m = 1$

$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$

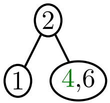
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



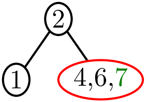
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



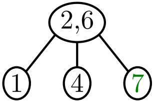
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



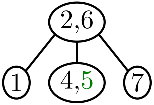
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



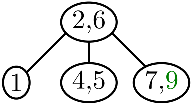
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



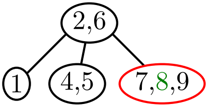
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



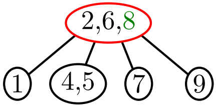
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



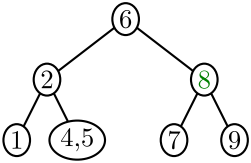
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



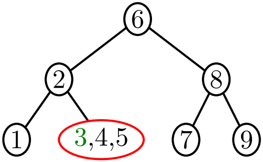
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



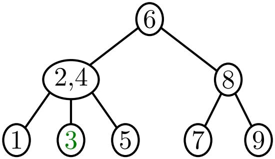
$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Insertion algorithm, $m = 1$



$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$

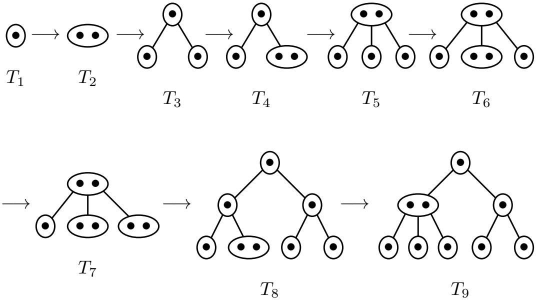
Insertion algorithm, $m = 1$



$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$

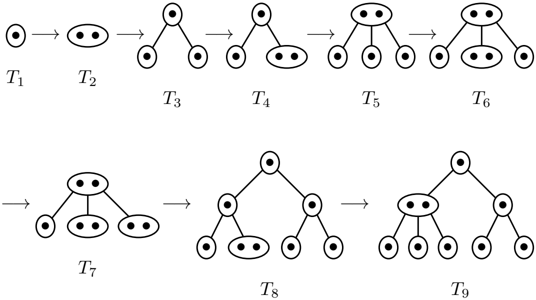
History of a B-tree

We are not interested in the exact value of the keys!



History of a B-tree

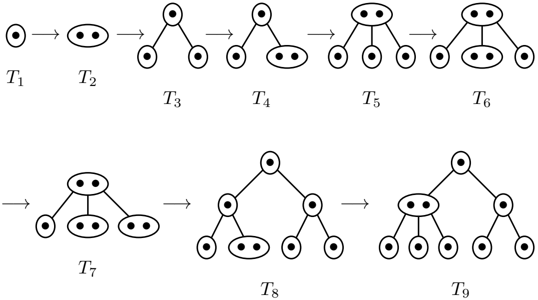
We are not interested in the exact value of the keys!



Such a sequence $(T_1, T_2, \ldots, T_n)$ is a *history* of $T = T_n$.

History of a B-tree

We are not interested in the exact value of the keys!



Such a sequence $(T_1, T_2, \ldots, T_n)$ is a *history* of $T = T_n$.
Set of all histories for B-trees with n keys: $\mathscr{H}_m(n)$

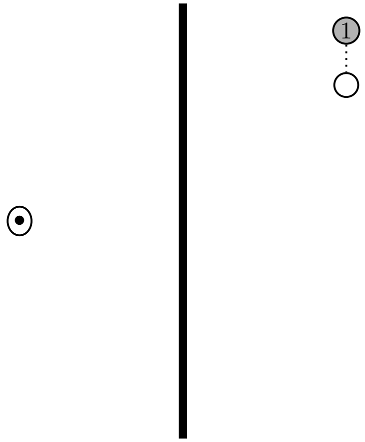
The bijection

Theorem

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:

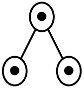
1. *H_n is a rooted plane tree on n vertices, labelled by $\{1, \ldots, n\}$, such that along each path from the root to a leaf, the labels are increasing.*
2. *The vertices of H_n at heights $2m, 3m+1, 4m+2, \ldots$ have up to two children, all other vertices have at most one child.*

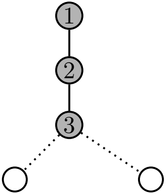
The bijection



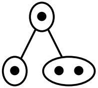
The bijection

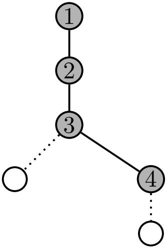
The bijection



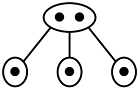


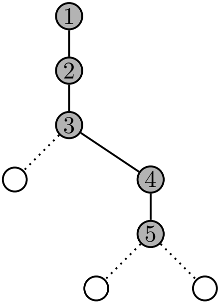
The bijection



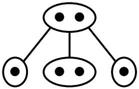


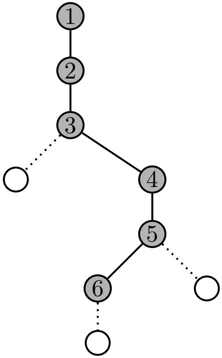
The bijection



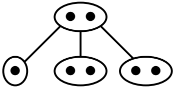


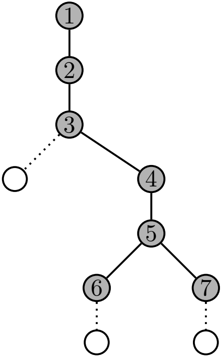
The bijection



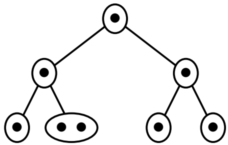


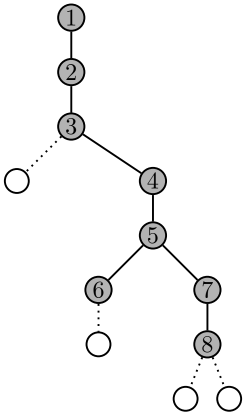
The bijection



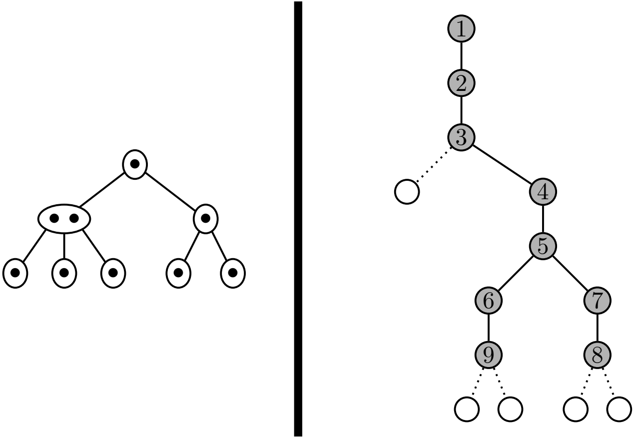


The bijection





The bijection



The bijection

Proposition

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

1. *For any $n \geq 1$, the number of external vertices of H_n equals the number of leaves of T_n.*
2. *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.*
3. *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.*

A consequence

$$\underline{\pi}(H_n) := \{\pi \in S_n : \pi \text{ yields the history belonging to } H_n\}$$

A consequence

$\underline{\pi}(H_n) := \{\pi \in S_n : \pi$ yields the history belonging to $H_n\}$

Proposition

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

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

Counting histories

For convenience, remove the first m vertices from a historic tree $\rightsquigarrow$ *reduced historic tree*.

Counting histories

For convenience, remove the first m vertices from a historic tree $\rightsquigarrow$ *reduced historic tree*.

This is amenable to analytic combinatorics:

$$H(x) = \sum_{n \geq 0} \frac{h_n}{n!} x^n$$

where h_n is the number of reduced historic trees with n internal vertices.

Counting histories

For convenience, remove the first m vertices from a historic tree $\rightsquigarrow$ *reduced historic tree*.
This is amenable to analytic combinatorics:

$$H(x) = \sum_{n \geq 0} \frac{h_n}{n!} x^n$$

where h_n is the number of reduced historic trees with n internal vertices.
For $m = 1$: Recursive structure gives

$$H''(x) = H(x)^2 \qquad H(0) = H'(0) = 1.$$

Counting histories

For convenience, remove the first m vertices from a historic tree $\rightsquigarrow$ *reduced historic tree*.

This is amenable to analytic combinatorics:

$$H(x) = \sum_{n \geq 0} \frac{h_n}{n!} x^n$$

where h_n is the number of reduced historic trees with n internal vertices.

For $m = 1$: Recursive structure gives

$$H''(x) = H(x)^2 \qquad H(0) = H'(0) = 1.$$

For general m:

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

Counting histories, $m = 1$

$$H''(x) = H(x)^2 \qquad H(0) = H'(0) = 1$$

Counting histories, $m = 1$

$$H''(x) = H(x)^2 \qquad H(0) = H'(0) = 1$$

This was already analysed (Bodini, Dien, Fontaine, Genitrini, Hwang, 2016) and has explicit solutions using the Weierstrass elliptic function: Dominant singularity at $\rho \approx 2.3758705509$ of order 2. Hence

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

Counting histories, general m

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

Counting histories, general m

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

DE not explicitly solvable. IF $H(x)$ has a dominant singularity at ρ_m, then

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

We conjecture that this is indeed true.

Counting histories, general m

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

DE not explicitly solvable. IF $H(x)$ has a dominant singularity at ρ_m, then

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

We conjecture that this is indeed true.
Numerical computations suggest

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

Tree statistics, $m = 1$

We can weigh historic trees H_n by $|\underline{\pi}(H_n)|$. Weighted e.g.f. for reduced historic trees satisfies

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

Tree statistics, $m = 1$

We can weigh historic trees H_n by $|\underline{\pi}(H_n)|$. Weighted e.g.f. for reduced historic trees satisfies

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

Explicit solution:

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

Tree statistics, $m = 1$

We can weigh historic trees H_n by $|\underline{\pi}(H_n)|$. Weighted e.g.f. for reduced historic trees satisfies

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

Explicit solution:

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

We can also include the number of external vertices $\mathrm{e}(T)$:

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

Tree statistics, $m = 1$

With this bivariate e.g.f.:

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

Tree statistics, $m = 1$

With this bivariate e.g.f.:

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

With singularity analysis + quasi-power theorem: CLT for $\mathrm{e}(T)$.
Moments can be computed via $W_1(x) = \frac{\partial}{\partial u} W(x,u)\Big|_{u=1}$.

Tree statistics, $m = 1$

With this bivariate e.g.f.:

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

With singularity analysis + quasi-power theorem: CLT for $\mathrm{e}(T)$. Moments can be computed via $W_1(x) = \frac{\partial}{\partial u} W(x,u)\Big|_{u=1}$.

Theorem

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 $\mathrm{Var}(L_n) = \frac{12}{637}(n+1)$ for $n > 11$. Moreover, the central limit theorem

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

holds.

Sets of permutations

Aim: Given a B-tree T, how do we get

$$\underline{\pi}(T) = \{\pi \in S_n : \pi \text{ yields } T\}?$$

(And also $\underline{\pi}(H_n)$)

Sets of permutations

Aim: Given a B-tree T, how do we get

$$\underline{\pi}(T) = \{\pi \in S_n : \pi \text{ yields } T\}?$$

(And also $\underline{\pi}(H_n)$)
Idea: If T has height 0, then $\underline{\pi}(T) = S_n$. Recurse over height.

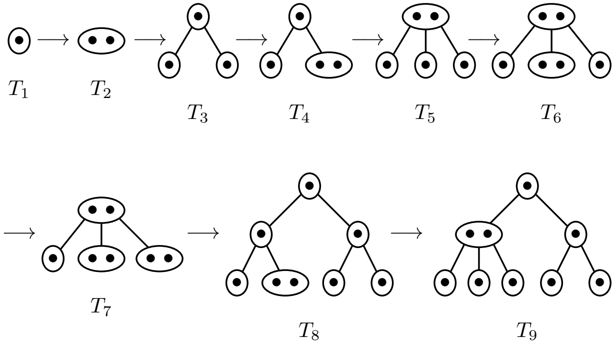
Sets of permutations

Aim: Given a B-tree T, how do we get

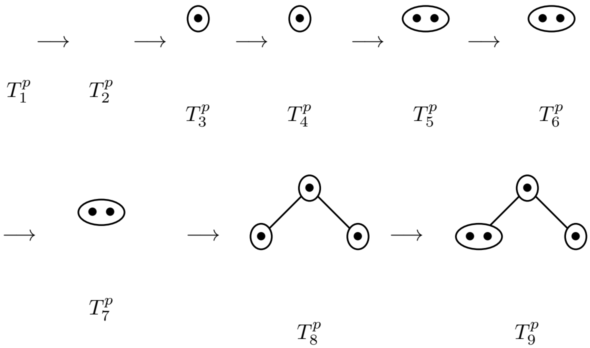
$$\underline{\pi}(T) = \{\pi \in S_n : \pi \text{ yields } T\}?$$

(And also $\underline{\pi}(H_n)$)
Idea: If T has height 0, then $\underline{\pi}(T) = S_n$. Recurse over height.
Why could this work?

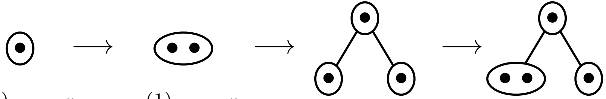
Pruning *B*-trees



Pruning B-trees



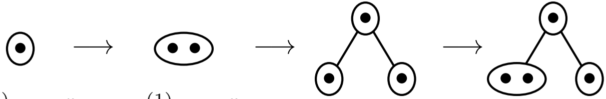
Pruning *B*-trees



$T_1^{(1)} = T_3^p$ $\quad$ $T_2^{(1)} = T_5^p$ $\quad$ $T_3^{(1)} = T_8^p$ $\quad$ $T_4^{(1)} = T_9^p$

This looks again like a history of a (smaller) *B*-tree!

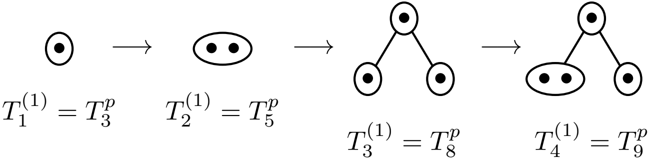
Pruning B-trees



$T_1^{(1)} = T_3^p$ $\quad$ $T_2^{(1)} = T_5^p$ $\quad$ $T_3^{(1)} = T_8^p$ $\quad$ $T_4^{(1)} = T_9^p$

This looks again like a history of a (smaller) B-tree!
Therefore, there is a permutation $\pi^{(1)}$ that produced this history...

Pruning B-trees



This looks again like a history of a (smaller) B-tree!
Therefore, there is a permutation $\pi^{(1)}$ that produced this history...
...and in fact, we can obtain $\pi^{(1)}$ exactly from the history of T.
Here, $\pi^{(1)} = (1, 3, 4, 2)$.

Algorithm: Overview

Given: $T = T_n$, and $\pi^{(1)}$.
Three steps:

Algorithm: Overview

Given: $T = T_n$, and $\pi^{(1)}$.

Three steps:

1. Take $\pi^{(1)}$ and T, and lift $\pi^{(1)}$ to a sequence $(K_{i_1}, \ldots, K_{i_{n_1}})$ of keys from T.

Algorithm: Overview

Given: $T = T_n$, and $\pi^{(1)}$.

Three steps:

1. Take $\pi^{(1)}$ and T, and lift $\pi^{(1)}$ to a sequence $(K_{i_1}, \ldots, K_{i_{n_1}})$ of keys from T.
2. Given $\pi^{(1)}$ and T, produce a historic tree H such that the pruned history fits with $\pi^{(1)}$.

Algorithm: Overview

Given: $T = T_n$, and $\pi^{(1)}$.

Three steps:

1. Take $\pi^{(1)}$ and T, and lift $\pi^{(1)}$ to a sequence $(K_{i_1}, \ldots, K_{i_{n_1}})$ of keys from T.
2. Given $\pi^{(1)}$ and T, produce a historic tree H such that the pruned history fits with $\pi^{(1)}$.
3. Given T, H, $(K_{i_1}, \ldots, K_{i_{n_1}})$, produce $\pi \in \underline{\pi}(H)$.

Algorithm: Overview

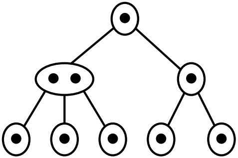
Given: $T = T_n$, and $\pi^{(1)}$.

Three steps:

1. Take $\pi^{(1)}$ and T, and lift $\pi^{(1)}$ to a sequence $(K_{i_1}, \ldots, K_{i_{n_1}})$ of keys from T.
2. Given $\pi^{(1)}$ and T, produce a historic tree H such that the pruned history fits with $\pi^{(1)}$.
3. Given T, H, $(K_{i_1}, \ldots, K_{i_{n_1}})$, produce $\pi \in \underline{\pi}(H)$.

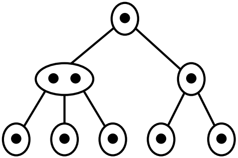
The algorithm requires choices, different choices lead to different π; all possibly choices = all permutations that produce T and respect $\pi^{(1)}$.

Algorithm: Step 1



$$\pi^{(1)} = (1, 3, 4, 2)$$

Algorithm: Step 1

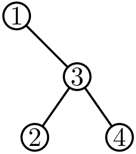


$$\pi^{(1)} = (1, 3, 4, 2)$$

In-order traversal reveals that the keys corresponding to $\pi^{(1)}$ are $(2, 6, 8, 4)$.

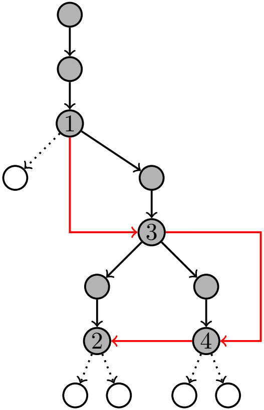
Algorithm: Step 2

Build a binary search tree from $\pi^{(1)}$:



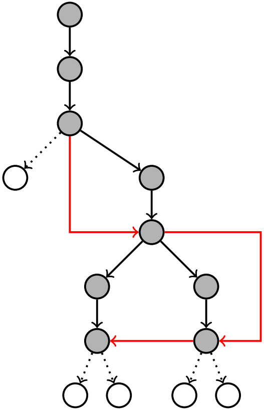
Algorithm: Step 2

Stretch it into a historic tree, and keep track of the insertion order into the binary search tree:



Algorithm: Step 2

Delete all labels:



Algorithm: Step 2

Lemma

The digraph $G = G(T, \pi^{(1)})$ constructed in this fashion is acyclic. Any topological labelling of G induces a historic tree H for T on the black edges. Such H corresponds bijectively to a history of T that is obtained by all those $\pi \in S_n$ that after pruning, produce $\pi^{(1)}$.

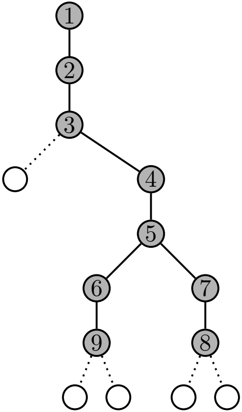
Algorithm: Step 2

Lemma

The digraph $G = G(T, \pi^{(1)})$ constructed in this fashion is acyclic. Any topological labelling of G induces a historic tree H for T on the black edges. Such H corresponds bijectively to a history of T that is obtained by all those $\pi \in S_n$ that after pruning, produce $\pi^{(1)}$.

We will choose the labelling we already know ;-)

Algorithm: Step 3

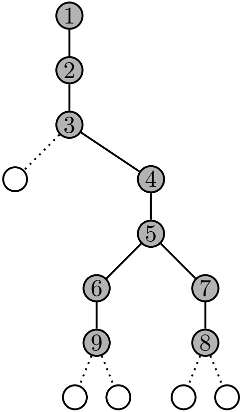


$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$

$\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\pi = (\_, \_, \_, \_, \_, \_, \_, \_, \_)$

Algorithm: Step 3

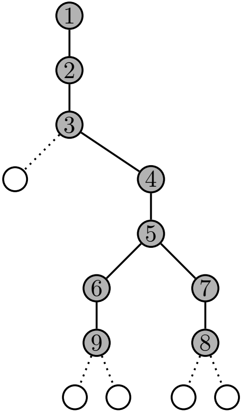


$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$

$\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

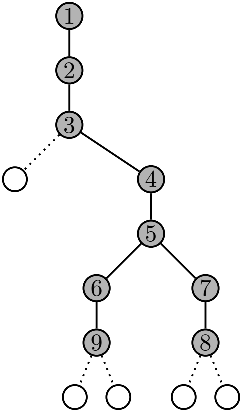
$\pi = (\ell, s, 2, \ell, \ell, \ell, \ell, \ell, \ell)$

Algorithm: Step 3



$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$
$\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
$\pi = (\ell, s, 2, \ell, \ell, \ell, \ell, \ell, \ell)$
This splits $\mathcal{R}$ and π:
$\mathcal{R}_- = \{1\}, \ \mathcal{R}_+ = \{3, 4, 5, 6, 7, 8, 9\}$
$\pi_- = (\_), \ \pi_+ = (\_, \_, \_, \_, \_, \_, \_)$

Algorithm: Step 3



$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$

$\mathcal{R} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$

$\pi = (\ell, 1, 2, \ell, \ell, \ell, \ell, \ell, \ell)$

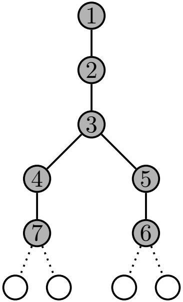
This splits $\mathcal{R}$ and π:

$\mathcal{R}_- = \{1\}, \ \mathcal{R}_+ = \{3, 4, 5, 6, 7, 8, 9\}$

$\pi_- = (1), \ \pi_+ = (\_, \_, \_, \_, \_, \_, \_)$

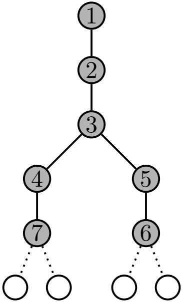
Let's follow along with π_+; update H.

Algorithm: Step 3



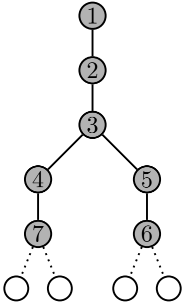
$$\pi^{(1)} = (1, 3, 4, 2) \mathrel{\hat{=}} (2, 6, 8, 4)$$
$$\mathcal{R} = \{3, 4, 5, 6, 7, 8, 9\}$$
$$\pi = (\_, 1, 2, \_, \_, \_, \_, \_, \_)$$
$$\mathrel{\hat{=}} (\_, \_, \_, \_, \_, \_, \_)$$

Algorithm: Step 3



$$\pi^{(1)} = (1,3,4,2) \widehat{=} (2,6,8,4)$$
$$\mathcal{R} = \{3,4,5,6,7,8,9\}$$
$$\pi = (6,1,2,\_,\_,\_,\_,\_,\_)$$
$$\widehat{=} (6,s,\ell,s,\ell,\ell,s)$$

Algorithm: Step 3



$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$

$\mathcal{R} = \{3, 4, 5, 6, 7, 8, 9\}$

$\pi = (6, 1, 2, \_, \_, \_, \_, \_, \_)$

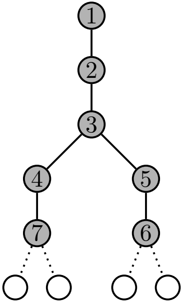
$\hat{=} (6, s, \ell, s, \ell, \ell, s)$

This splits $\mathcal{R}$ and π:

$\mathcal{R}_- = \{3, 4, 5\}, \ \mathcal{R}_+ = \{7, 8, 9\}$

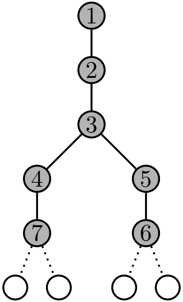
$\pi_- = (\_, \_, \_), \ \pi_+ = (\_, \_, \_)$

Algorithm: Step 3



$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$
$\pi = (6, 1, 2, 4, \_, 5, \_, \_, 3)$
$\hat{=} (6, s, \ell, s, \ell, \ell, s)$
Anything goes for $\pi_\pm$:
$\mathcal{R}_- = \{3, 4, 5\}, \ \mathcal{R}_+ = \{7, 8, 9\}$
$\pi_- = (4, 5, 3), \ \pi_+ = (7, 9, 8)$

Algorithm: Step 3



$\pi^{(1)} = (1, 3, 4, 2) \hat{=} (2, 6, 8, 4)$

$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$

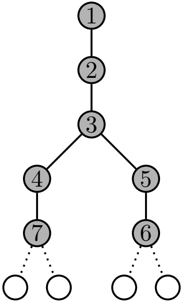
$\quad \hat{=} (6, 4, \ell, 5, \ell, \ell, 3)$

Anything goes for $\pi_\pm$:

$\mathcal{R}_- = \{3, 4, 5\},\ \mathcal{R}_+ = \{7, 8, 9\}$

$\pi_- = (4, 5, 3),\ \pi_+ = (7, 9, 8)$

Algorithm: Step 3



$$\pi = (6, 1, 2, 4, 7, 5, 9, 8, 3)$$

Conclusions and outlook

- Bijection between histories of B-trees and family of increasing trees.

Conclusions and outlook

- Bijection between histories of B-trees and family of increasing trees.
- Enables new approaches for counting B-trees and for showing limit theorems for tree statistics.

Conclusions and outlook

- Bijection between histories of B-trees and family of increasing trees.
- Enables new approaches for counting B-trees and for showing limit theorems for tree statistics.
- Algorithmic description of set of permutations that produce given B-tree or given history.

Conclusions and outlook

- Bijection between histories of B-trees and family of increasing trees.
- Enables new approaches for counting B-trees and for showing limit theorems for tree statistics.
- Algorithmic description of set of permutations that produce given B-tree or given history.
- Many open problems remain!

Conclusions and outlook

- Bijection between histories of B-trees and family of increasing trees.
- Enables new approaches for counting B-trees and for showing limit theorems for tree statistics.
- Algorithmic description of set of permutations that produce given B-tree or given history.
- Many open problems remain!

∼∼∼ $\mathscr{F}\mathscr{I}\mathscr{N}$ ∼∼∼