Dependencies between Shape Parameters in Random Log-Trees

(joint with H.-H. Chern, H.-K. Hwang and R. Neininger)

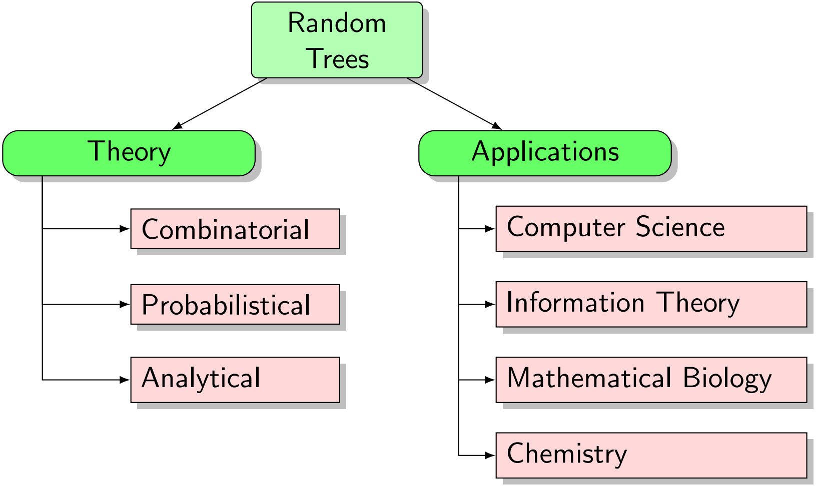
Michael Fuchs

Institute of Applied Mathematics
National Chiao Tung University

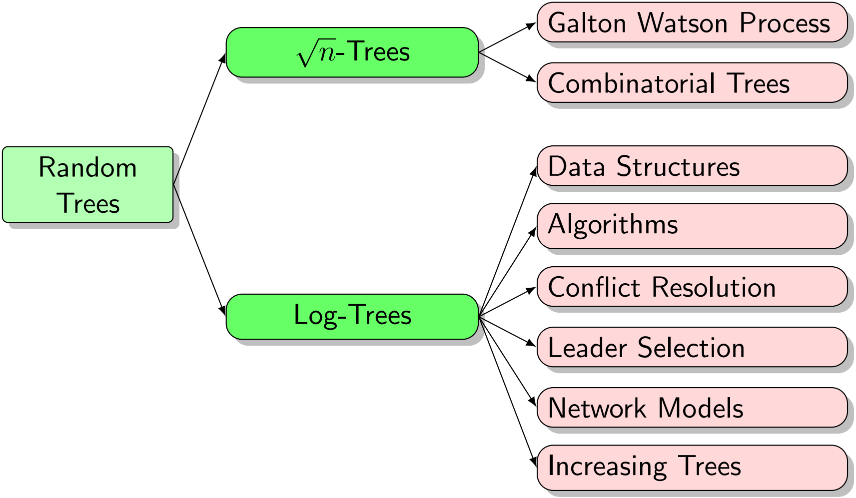
Hsinchu, Taiwan

Chennai, July 10, 2015

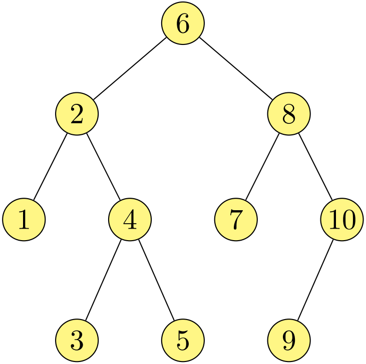
Random Trees



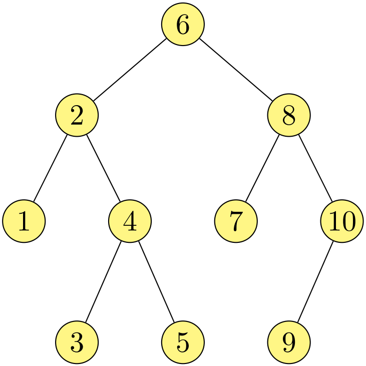
$\sqrt{n}$-Trees vs. Log-Trees



Random Log-Trees



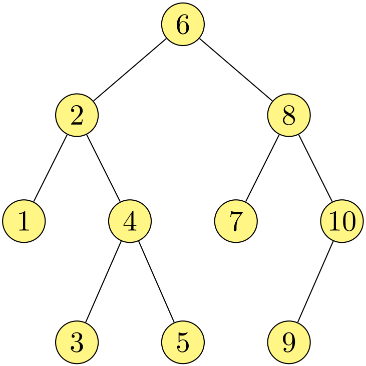
Random Log-Trees



Trees are equipped with a random model

$\longrightarrow$ **Random Trees**

Random Log-Trees



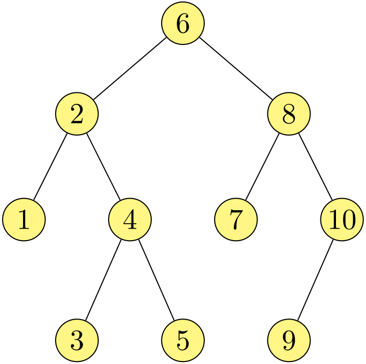
Trees are equipped with a random model

$\longrightarrow$ **Random Trees**

Average height of logarithmic order

$\longrightarrow$ **Random Log-Trees**

Random Log-Trees



Trees are equipped with a random model

$\longrightarrow$ **Random Trees**

Average height of logarithmic order

$\longrightarrow$ **Random Log-Trees**

Properties are described via **Shape Parameters**

Examples of Random Log-Trees

- **Binary Search Trees and Variants**

  Binary search trees, m-ary search trees, fringe balanced binary search trees, quadtrees, simplex trees, etc.

Examples of Random Log-Trees

- **Binary Search Trees and Variants**

  Binary search trees, m-ary search trees, fringe balanced binary search trees, quadtrees, simplex trees, etc.

- **Digital Trees**

  Digital search trees, bucket digital search trees, tries, PATRICIA tries, suffix trees, etc.

Examples of Random Log-Trees

- **Binary Search Trees and Variants**

  Binary search trees, m-ary search trees, fringe balanced binary search trees, quadtrees, simplex trees, etc.

- **Digital Trees**

  Digital search trees, bucket digital search trees, tries, PATRICIA tries, suffix trees, etc.

- **Increasing Trees**

  Binary increasing trees (=binary search trees), recursive trees, plane-oriented recursive trees, etc.

Binary Search Trees (BSTs)

Input:

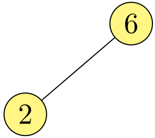
6, 2, 4, 8, 7, 1, 5, 3, 10, 9

Binary Search Trees (BSTs)

Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

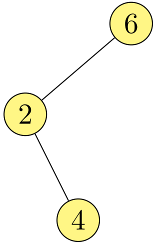
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

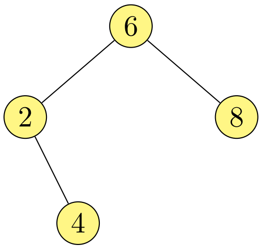
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

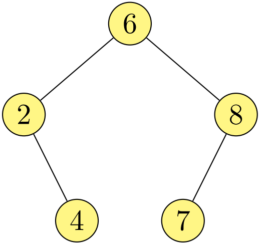
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

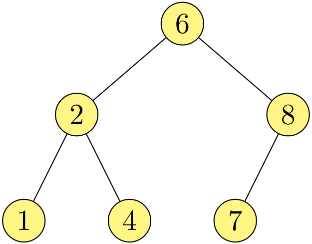
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

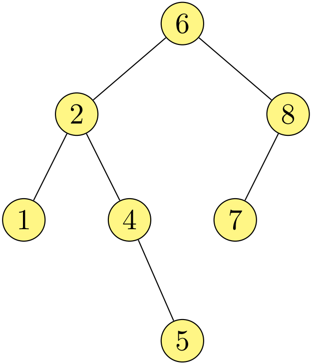
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

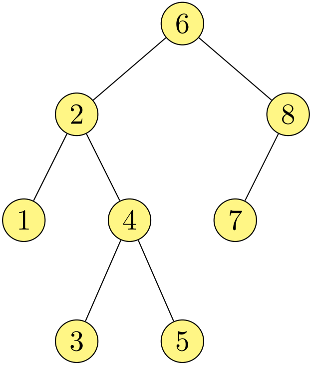
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

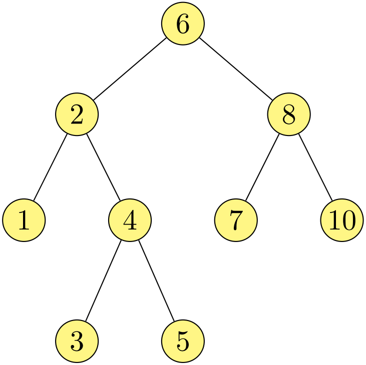
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

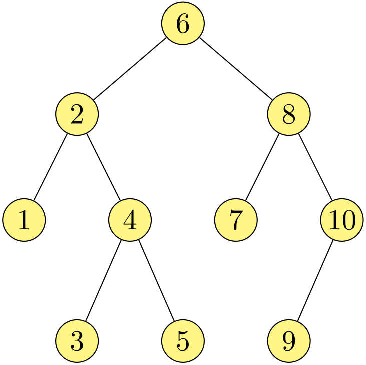
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

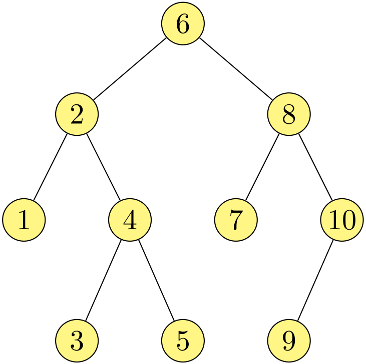
Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

Binary Search Trees (BSTs)



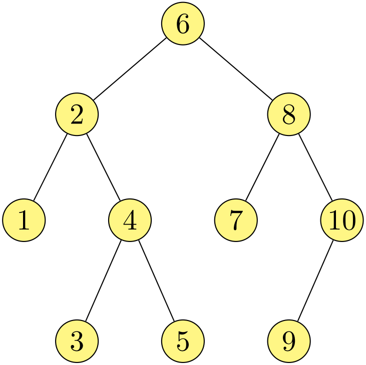
Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

If every permutation of the input sequence is equally likely

$\longrightarrow$ **Random BSTs**

Binary Search Trees (BSTs)



Input:

6, 2, 4, 8, 7, 1, 5, 3, 10, 9

If every permutation of the input sequence is equally likely

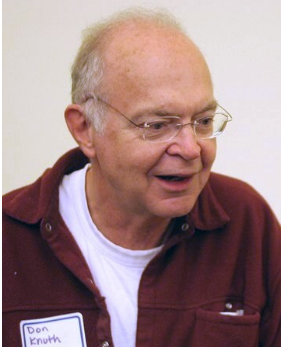
$\longrightarrow$ **Random BSTs**

Shape parameters become random variables

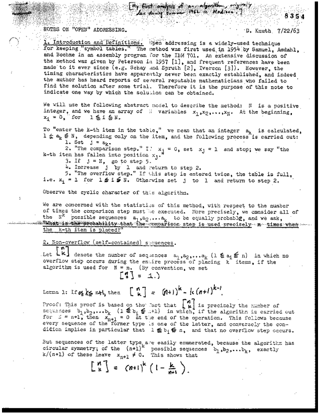
Examples of Shape Parameters

- **Height** (= maximal root-distance)
- **Depth** (= root-distance of a random node)
- **Total Path Length** (= sum of all root-distances)
- **Size or Storage Requirement**
- **Number of Leaves** (or more generally, number of nodes of fixed out-degree)
- **Patterns**
- **Profiles** (node profile, subtree size profile, etc.)

(Average Case) Analysis of Algorithms



Donald E. Knuth



[My first analysis of an algorithm, originally done during summer 1962 in Madison.]

8354

NOTES ON "OPEN" ADDRESSING. D. Knuth 7/22/63

1. Introduction and Definitions. Open addressing is a widely-used technique for keeping "symbol tables." The method was first used in 1954 by Samuel, Amdahl, and Boehme in an assembly program for the IBM 701. An extensive discussion of the method was given by Peterson in 1957 [1], and frequent references have been made to it ever since (e.g. Schay and Spruth [2], Iverson [3]). However, the timing characteristics have apparently never been exactly established, and indeed the author has heard reports of several reputable mathematicians who failed to find the solution after some trial. Therefore it is the purpose of this note to indicate one way by which the solution can be obtained.

We will use the following abstract model to describe the method: N is a positive integer, and we have an array of N variables $x_1, x_2, \ldots, x_N$. At the beginning, $x_i = 0$, for $1 \leq i \leq N$.

To "enter the k-th item in the table," we mean that an integer a_k is calculated, $1 \leq a_k \leq N$, depending only on the item, and the following process is carried out:

1. Set $j = a_k$.
2. "The comparison step." If $x_j = 0$, set $x_j = 1$ and stop; we say "the k-th item has fallen into position x_j."
3. If $j = N$, go to step 5.
4. Increase j by 1 and return to step 2.
5. "The overflow step." If this step is entered twice, the table is full, i.e. $x_i = 1$ for $1 \leq i \leq N$. Otherwise set j to 1 and return to step 2.

Observe the cyclic character of this algorithm.

We are concerned with the statistics of this method, with respect to the number of times the comparison step must be executed. More precisely, we consider all of the N^k possible sequences $a_1, a_2, \ldots a_k$ to be equally probable, and we ask, "What is the probability that the comparison step is used precisely m times when the k-th item is placed?"

2. Non-overflow (self-contained) sequences.

Let $\left[{n \atop k}\right]$ denote the number of sequences $a_1, a_2, \ldots a_k$ $(1 \leq a_i \leq n)$ in which no overflow step occurs during the entire process of placing k items, if the algorithm is used for $N = n$. (By convention, we set $\left[{n \atop 0}\right] = 1$.)

Lemma 1: If $0 \leq k \leq n+1$, then $\left[{n \atop k}\right] = (n+1)^k - k(n+1)^{k-1}$.

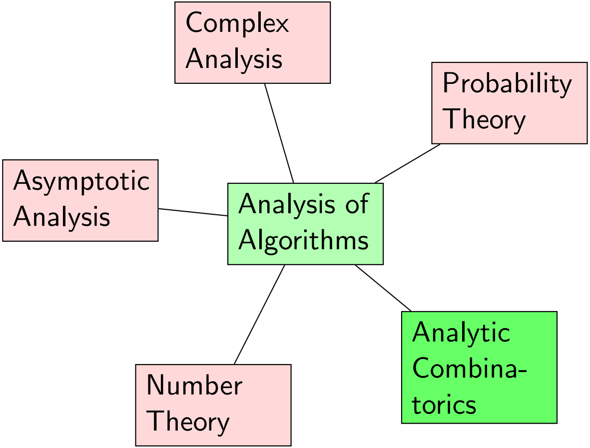
Proof: This proof is based on the fact that $\left[{n \atop k}\right]$ is precisely the number of sequences $b_1, b_2, \ldots b_k$ $(1 \leq b_i \leq n+1)$ in which, if the algorithm is carried out for $N = n+1$, then $x_{n+1} = 0$ at the end of the operation. This follows because every sequence of the former type is one of the latter, and conversely the condition implies in particular that $1 \leq b_i \leq n$, and that no overflow step occurs.

But sequences of the latter type are easily enumerated, because the algorithm has circular symmetry; of the $(n+1)^k$ possible sequences $b_1, b_2, \ldots b_k$, exactly $k/(n+1)$ of these leave $x_{n+1} \neq 0$. This shows that

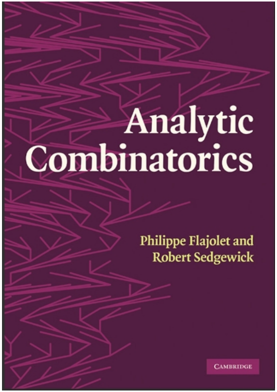
$$\left[{n \atop k}\right] = (n+1)^k \left(1 - \frac{k}{n+1}\right).$$

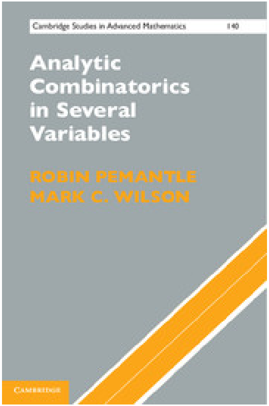
Notes on Open Addressing

Analysis of Algorithms and Related Fields



Analytic Combinatorics





Random m-ary Search Trees

Proposed by Muntz and Uzgalis in 1971.

Random m-ary Search Trees

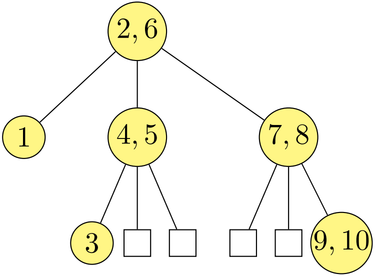
Proposed by Muntz and Uzgalis in 1971.

Input: 6, 2, 4, 8, 7, 1, 5, 3, 10, 9

Random m-ary Search Trees

Proposed by Muntz and Uzgalis in 1971.

Input: 6, 2, 4, 8, 7, 1, 5, 3, 10, 9

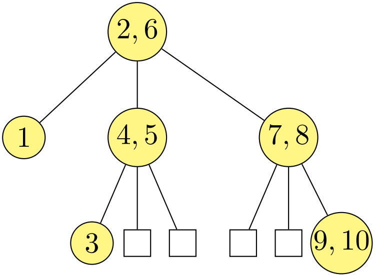


$m = 3$

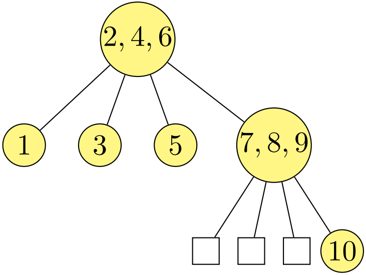
Random m-ary Search Trees

Proposed by Muntz and Uzgalis in 1971.

Input: 6, 2, 4, 8, 7, 1, 5, 3, 10, 9



$m = 3$

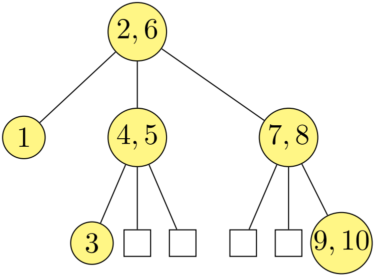


$m = 4$

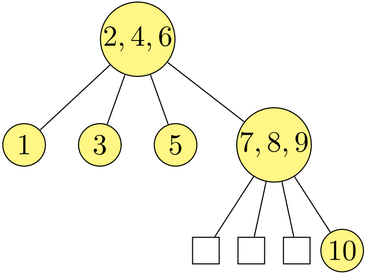
Random m-ary Search Trees

Proposed by Muntz and Uzgalis in 1971.

Input: 6, 2, 4, 8, 7, 1, 5, 3, 10, 9



$m = 3$



$m = 4$

If permutations are equally likely $\longrightarrow$ **random m-ary search trees**

Size, KPL, and NPL

- **Size** (or Storage Requirement)

  Number of nodes holding keys. Only random if $m \geq 2$.

  $S_n =$ size of a random m-ary search tree built from n keys.

Size, KPL, and NPL

- **Size** (or Storage Requirement)

  Number of nodes holding keys. Only random if $m \geq 2$.

  S_n = size of a random m-ary search tree built from n keys.

- **Key Path Length** (KPL)

  Sum of all key-distances to the root.

  K_n = KPL of a random m-ary search tree built from n keys.

Size, KPL, and NPL

- **Size** (or Storage Requirement)

  Number of nodes holding keys. Only random if $m \geq 2$.

  $S_n =$ size of a random m-ary search tree built from n keys.

- **Key Path Length** (KPL)

  Sum of all key-distances to the root.

  $K_n =$ KPL of a random m-ary search tree built from n keys.

- **Node Path Length** (NPL)

  Sum of all node-distances to the root.

  $N_n =$ NPL of a random m-ary search tree built from n keys.

Size: Mean

Knuth (1973):

$$\mathbb{E}(S_n) \sim \phi n,$$

where

$$\phi := \frac{1}{2(H_m - 1)}$$

and H_m are the Harmonic numbers.

Size: Mean

Knuth (1973):

$$\mathbb{E}(S_n) \sim \phi n,$$

where

$$\phi := \frac{1}{2(H_m - 1)}$$

and H_m are the Harmonic numbers.

Mahmoud and Pittel (1989):

$$\mathbb{E}(S_n) = \phi(n+1) - \frac{1}{m-1} + \mathcal{O}(n^{\alpha-1}),$$

where α is the real part of the second largest zero of

$$\Lambda(z) = z(z+1)\cdots(z+m-2) - m!.$$

Size: Phase Change for Variance

Mahmoud and Pittel (1989):

$$\mathrm{Var}(S_n) \sim \begin{cases} C_S n, & \text{if } m \le 26; \\ F_1(\beta \log n) n^{2\alpha-2}, & \text{if } m \ge 27, \end{cases}$$

where $\lambda = \alpha + i\beta$ is the second largest zero of $\Lambda(z)$.

Size: Phase Change for Variance

Mahmoud and Pittel (1989):

$$\mathrm{Var}(S_n) \sim \begin{cases} C_S n, & \text{if } m \le 26; \\ F_1(\beta \log n) n^{2\alpha-2}, & \text{if } m \ge 27, \end{cases}$$

where $\lambda = \alpha + i\beta$ is the second largest zero of $\Lambda(z)$.

Here, $F_1(z)$ is the periodic function

$$\begin{aligned} F_1(z) = 2\frac{|A|^2}{|\Gamma(\lambda)|^2} &\left(-1 + \frac{m!(m-1)|\Gamma(\lambda)|^2}{\Gamma(2\alpha+m-2) - m!\Gamma(2\alpha-1)}\right) \\ &+ 2\Re\left(\frac{A^2 e^{2iz}}{\Gamma(\lambda)^2}\left(-1 + \frac{m!(m-1)\Gamma(\lambda)^2}{\Gamma(2\lambda+m-2) - m!\Gamma(2\lambda-1)}\right)\right) \end{aligned}$$

with $A = 1/(\lambda(\lambda-1)\sum_{0\le j\le m-2}\frac{1}{j+\lambda})$.

Size: Phase Change for Limit Law

Theorem (Mahmoud & Pittel (1989); Lew & Mahmoud (1994))

For $3 \leq m \leq 26$,

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

where $N(0,1)$ *is the standard normal distribution.*

Size: Phase Change for Limit Law

Theorem (Mahmoud & Pittel (1989); Lew & Mahmoud (1994))

For $3 \leq m \leq 26$,

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

where $N(0,1)$ *is the standard normal distribution.*

Theorem (Chern & Hwang (2001))

For $m \geq 27$,

$$\frac{S_n - \mathbb{E}(S_n)}{\sqrt{\mathrm{Var}(S_n)}}$$

does not converge to a fixed limit law.

KPL: Moments

Mahmoud (1986):

$$\mathbb{E}(K_n) = 2\phi n \log n + c_1 n + o(n),$$

where c_1 is an explicitly computable constant.

KPL: Moments

Mahmoud (1986):

$$\mathbb{E}(K_n) = 2\phi n \log n + c_1 n + o(n),$$

where c_1 is an explicitly computable constant.

Mahmoud (1992):

$$\mathrm{Var}(K_n) \sim C_K n^2,$$

where

$$C_K = 4\phi^2 \left(\frac{(m+1)H_m^{(2)} - 2}{m-1} - \frac{\pi^2}{6} \right)$$

with $H_m^{(2)} = \sum_{1 \le j \le m} 1/j^2$.

KPL: Moments

Mahmoud (1986):

$$\mathbb{E}(K_n) = 2\phi n \log n + c_1 n + o(n),$$

where c_1 is an explicitly computable constant.

Mahmoud (1992):

$$\mathrm{Var}(K_n) \sim C_K n^2,$$

where

$$C_K = 4\phi^2 \left(\frac{(m+1)H_m^{(2)} - 2}{m-1} - \frac{\pi^2}{6} \right)$$

with $H_m^{(2)} = \sum_{1 \le j \le m} 1/j^2$.

So, no phase change here for the variance!

KPL: Limit Law

Theorem (Neininger & Rüschendorf (1999))

We have,

$$\frac{K_n - \mathbb{E}(K_n)}{n} \xrightarrow{d} K,$$

where K is the unique solution of

$$X \stackrel{d}{=} \sum_{1 \le r \le m} V_r X^{(r)} + 2\phi \sum_{1 \le r \le m} V_r \log V_r$$

with $X^{(r)}$ an independent copy of X and

$$V_r = U_{(r)} - U_{(r-1)},$$

where $U_{(r)}$ is the r-th order statistic of m i.i.d. uniform RVs.

Node Path Length (NPL)

N_n = sum of all node-distances in an m-search tree built from n keys.

Node Path Length (NPL)

N_n = sum of all node-distances in an m-search tree built from n keys.

Broutin and Holmgren (2012):

$$\mathbb{E}(N_n) = 2\phi^2 n \log n + c_2 n + o(n),$$

where c_2 is an explicitly computable constant.

Node Path Length (NPL)

N_n = sum of all node-distances in an m-search tree built from n keys.

Broutin and Holmgren (2012):

$$\mathbb{E}(N_n) = 2\phi^2 n \log n + c_2 n + o(n),$$

where c_2 is an explicitly computable constant.

We have,

$$\begin{cases} S_n \stackrel{d}{=} S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)} + 1, \\ N_n \stackrel{d}{=} N_{I_1}^{(1)} + \cdots + N_{I_m}^{(m)} + S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)}. \end{cases}$$

Node Path Length (NPL)

N_n = sum of all node-distances in an m-search tree built from n keys.

Broutin and Holmgren (2012):

$$\mathbb{E}(N_n) = 2\phi^2 n \log n + c_2 n + o(n),$$

where c_2 is an explicitly computable constant.

We have,

$$\begin{cases} S_n \stackrel{d}{=} S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)} + 1, \\ N_n \stackrel{d}{=} N_{I_1}^{(1)} + \cdots + N_{I_m}^{(m)} + S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)}. \end{cases}$$

So, one expects a strong positive dependence between S_n and N_n!

Size and NPL: Correlation (i)

Theorem (Chern, F., Hwang, Neininger (2015+))

We have,

$$\mathrm{Cov}(S_n, N_n) \sim \begin{cases} C_R n \log n, & \text{if } 3 \le m \le 13; \\ \phi F_2(\beta \log n) n^{\alpha}, & \text{if } m \ge 14, \end{cases},$$

where C_R *is a constant and* $F_2(z)$ *is a periodic function. Moreover,*

$$\mathrm{Var}(N_n) \sim \phi^2 C_K n^2.$$

Size and NPL: Correlation (i)

Theorem (Chern, F., Hwang, Neininger (2015+))

We have,

$$\mathrm{Cov}(S_n, N_n) \sim \begin{cases} C_R n \log n, & \text{if } 3 \le m \le 13; \\ \phi F_2(\beta \log n) n^{\alpha}, & \text{if } m \ge 14, \end{cases},$$

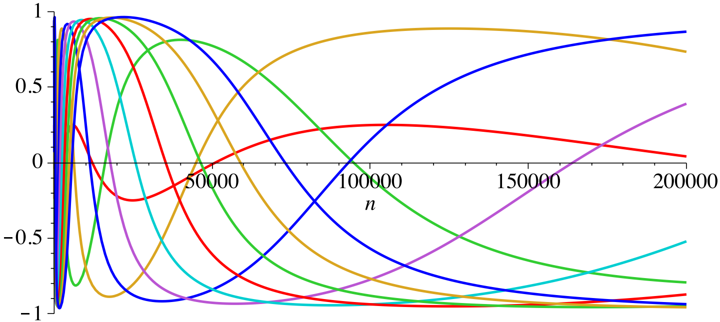
where C_R *is a constant and* $F_2(z)$ *is a periodic function. Moreover,*

$$\mathrm{Var}(N_n) \sim \phi^2 C_K n^2.$$

Thus (!),

$$\rho(S_n, N_n) \begin{cases} \longrightarrow 0, & \text{if } 3 \le m \le 26; \\ \sim \frac{F_2(\beta \log n)}{\sqrt{C_K F_1(\beta \log n)}}, & \text{if } m \ge 27. \end{cases}$$

Size and NPL: Correlation (ii)



Periodic function of $\rho(S_n, N_n)$ for $m = 27, 54, \ldots, 270$.

Pearson's Correlation Coefficient

Pearson: for RVs X and Y

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

Measures linear dependence between X and Y!

Pearson's Correlation Coefficient

Pearson: for RVs X and Y

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

Measures linear dependence between X and Y!

Refined correlation measures:

Distance correlation, Brownian covariance, mutual information, total correlation, dual total correlation, etc.

Pearson's Correlation Coefficient

Pearson: for RVs X and Y

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

Measures linear dependence between X and Y!

Refined correlation measures:

Distance correlation, Brownian covariance, mutual information, total correlation, dual total correlation, etc.

Question: Can our counterintuitive result for $\rho(S_n, N_n)$ be ascribed to the weakness of Pearson's correlation coefficient?

Pearson's Correlation Coefficient

Pearson: for RVs X and Y

$$\rho(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}.$$

Measures linear dependence between X and Y!

Refined correlation measures:

Distance correlation, Brownian covariance, mutual information, total correlation, dual total correlation, etc.

Question: Can our counterintuitive result for $\rho(S_n, N_n)$ be ascribed to the weakness of Pearson's correlation coefficient? NO!

Size and NPL: Limit Law for $3 \leq m \leq 26$

Theorem (Chern, F., Hwang, Neininger (2015+))

Consider

$$Q_n = (S_n, N_n).$$

Then,

$$\mathrm{Cov}(Q_n)^{-1/2}(Q_n - \mathbb{E}(Q_n)) \xrightarrow{d} (N, K),$$

where N has a standard normal distribution.

Moreover, N and K are independent!

Size and NPL: Limit Law for $3 \leq m \leq 26$

Theorem (Chern, F., Hwang, Neininger (2015+))

Consider

$$Q_n = (S_n, N_n).$$

Then,

$$\mathrm{Cov}(Q_n)^{-1/2}(Q_n - \mathbb{E}(Q_n)) \xrightarrow{d} (N, K),$$

where N has a standard normal distribution.

Moreover, N and K are independent!

Thus, asymptotic independence for $3 \leq m \leq 26$ is also observed in the bivariate limit law!

Size and NPL: Limit Law for $m \geq 27$

Theorem (Chern, F., Hwang, Neininger (2015+))

Consider

$$Y_n = \left(\frac{S_n - \phi n}{n^{\alpha-1}}, \frac{N_n - \mathbb{E}(N_n)}{n} \right).$$

Then,

$$\ell_2(Y_n, (\Re(n^{i\beta}\Lambda), K)) \longrightarrow 0,$$

where ℓ_2 *is the minimal* L_2*-metric and* Λ *is the unique solution of*

$$W \stackrel{d}{=} \sum_{1\leq r\leq m} V_r^{\lambda-1} W^{(r)}$$

with $W^{(r)}$ *independent copies of* W*.*

Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t + 1$ reorganized such that the median becomes the root.

Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t+1$ reorganized such that the median becomes the root.

Example: $t = 1$ and input sequence $3, 1, 2$

Fringe Balanced Binary Search Trees (FBBSTs)

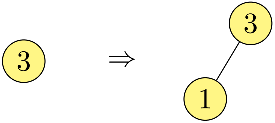
Constructed like a binary search tree with every subtree of size $2t+1$ reorganized such that the median becomes the root.

Example: $t=1$ and input sequence $3,1,2$

Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t + 1$ reorganized such that the median becomes the root.

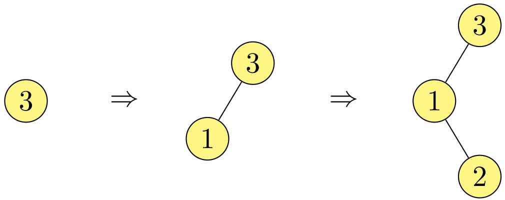
Example: $t = 1$ and input sequence $3, 1, 2$



Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t+1$ reorganized such that the median becomes the root.

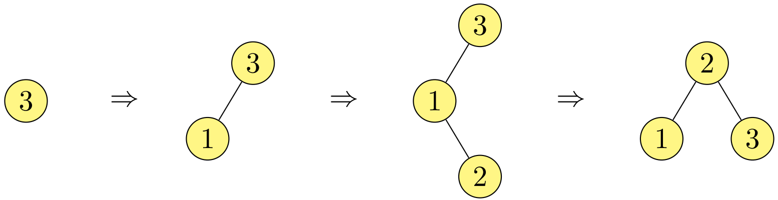
Example: $t = 1$ and input sequence $3, 1, 2$



Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t+1$ reorganized such that the median becomes the root.

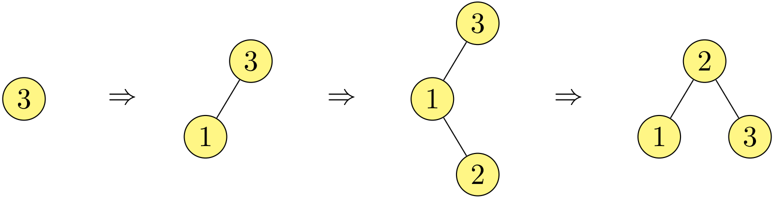
Example: $t = 1$ and input sequence $3, 1, 2$



Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t+1$ reorganized such that the median becomes the root.

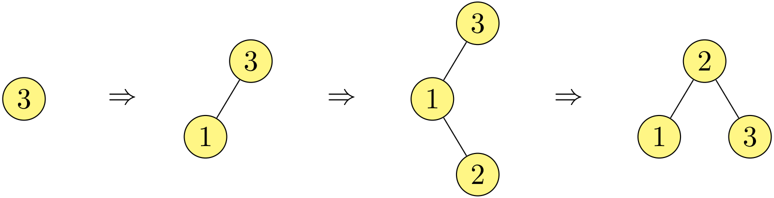
Example: $t = 1$ and input sequence $3, 1, 2$



Fringe Balanced Binary Search Trees (FBBSTs)

Constructed like a binary search tree with every subtree of size $2t + 1$ reorganized such that the median becomes the root.

Example: $t = 1$ and input sequence $3, 1, 2$



S_n = number of nodes with subtrees of size at least $2t + 1$.

T_n = root-distances of nodes with subtrees of size at least $2t + 1$.

FBBSTs: Means

Chern and Hwang (2001):

$$\mathbb{E}(S_n) = \frac{n+1}{2(t+1)(H_{2t+2} - H_{t+1})} - 1 + \mathcal{O}(n^{\alpha_t - 1}),$$

where α_t is the real part of the second largest zero of

$$\Lambda_t(z) = (z+t)\cdots(z+2t) - \frac{2(2t+1)!}{t!}.$$

FBBSTs: Means

Chern and Hwang (2001):

$$\mathbb{E}(S_n) = \frac{n+1}{2(t+1)(H_{2t+2} - H_{t+1})} - 1 + \mathcal{O}(n^{\alpha_t - 1}),$$

where α_t is the real part of the second largest zero of

$$\Lambda_t(z) = (z+t)\cdots(z+2t) - \frac{2(2t+1)!}{t!}.$$

With the tools from Chern and Hwang (2001):

$$\mathbb{E}(T_n) = \frac{n \log n}{H_{2t+2} - H_{t+1}} + c_t n + o(n),$$

where c_t is an explicitly computable constant.

FBBSTs: Variances and Covariance

Theorem (Chern, F., Hwang, Neininger (2015+))

We have,

$$\mathrm{Var}(S_n) \sim \begin{cases} D_S n, & \text{if } 1 \le t \le 58; \\ G_1(\beta_t \log n) n^{2\alpha_t - 2}, & \text{if } t \ge 59, \end{cases}$$

$$\mathrm{Cov}(S_n, T_n) \sim \begin{cases} D_R n, & \text{if } 1 \le t \le 28; \\ G_2(\beta_t \log n) n^{\alpha_t}, & \text{if } t \ge 29, \end{cases}$$

$$\mathrm{Var}(T_n) \sim D_T n^2,$$

where D_S, D_R, D_T are constants and $G_1(z), G_2(z)$ are periodic functions. Moreover, $\lambda_t = \alpha_t + i\beta_t$ is the second largest root of $\Lambda_t(z)$.

FBBSTs: Limit Law for $1 \le t \le 58$

Theorem (Chern, F., Hwang, Neininger (2015+))

For $X_n = (S_n, T_n)$, we have

$$\mathrm{Cov}(X_n)^{-1/2}(X_n - \mathbb{E}(X_n)) \xrightarrow{d} (N, T),$$

with N, T independent, where N has a standard normal distribution and T is the unique solution of

$$X \stackrel{d}{=} VX^{(1)} + (1-V)X^{(2)} + D_X^{-1/2} + \frac{1}{D_X^{1/2}(H_{2t+2} - H_{t+1})}(V\log V + (1-V)\log(1-V)),$$

where $X^{(i)}$ are independent copies of X and V is the median of $2t+1$ i.i.d. uniform RVs.

FBBSTs: Limit Law for $t \geq 59$

Theorem (Chern, F., Hwang, Neininger (2015+))

Consider

$$Z_n = \left(\frac{S_n - n/((t+1)(H_{2t+2} - H_{t+1}))}{n^{\alpha_t - 1}}, \frac{T_n - \mathbb{E}(T_n)}{n} \right).$$

Then,

$$\ell_2(Z_n, (\Re(n^{i\beta}\Lambda), T)) \longrightarrow 0,$$

where Λ is the unique solution of

$$W \stackrel{d}{=} V^{\lambda_t} W^{(1)} + (1-V)^{\lambda_t} W^{(2)}$$

with $W^{(i)}$ independent copies of W.

Median-of-$2t+1$ Quicksort

Invented by T. Hoare in 1960.

Median-of-$2t + 1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Median-of-$2t+1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Example: $t = 0$ and input sequence $3, 1, 5, 6, 2, 4$.

Median-of-$2t + 1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Example: $t = 0$ and input sequence $3, 1, 5, 6, 2, 4$.

(i) Choose first key (3) as pivot element.

Median-of-$2t + 1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Example: $t = 0$ and input sequence $3, 1, 5, 6, 2, 4$.

(i) Choose first key (3) as pivot element.

(ii) Split the remaining keys into two sequences, one containing all keys smaller than the pivot $(1, 2)$ and the other containing all keys larger than the pivot $(5, 6, 4)$.

Median-of-$2t+1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Example: $t = 0$ and input sequence $3, 1, 5, 6, 2, 4$.

(i) Choose first key (3) as pivot element.

(ii) Split the remaining keys into two sequences, one containing all keys smaller than the pivot $(1, 2)$ and the other containing all keys larger than the pivot $(5, 6, 4)$.

(iii) Recursively continue with the subsequences.

Median-of-$2t + 1$ Quicksort

Invented by T. Hoare in 1960.

One of the most important algorithm in computer science.

Example: $t = 0$ and input sequence $3, 1, 5, 6, 2, 4$.

(i) Choose first key (3) as pivot element.

(ii) Split the remaining keys into two sequences, one containing all keys smaller than the pivot $(1, 2)$ and the other containing all keys larger than the pivot $(5, 6, 4)$.

(iii) Recursively continue with the subsequences.

Median of $2t + 1$ keys as pivot $\longrightarrow$ **Median-of-$2t + 1$ Quicksort**

Comparisons and Partitioning Stages

Consider quicksort on a random permutation of length n.

Comparisons and Partitioning Stages

Consider quicksort on a random permutation of length n.

$C_n =$ number of key comparison.

$P_n =$ number of recursive calls (=partitioning stages).

Comparisons and Partitioning Stages

Consider quicksort on a random permutation of length n.

C_n = number of key comparison.

P_n = number of recursive calls (=partitioning stages).

Median-of-$2t+1$ Quicksort $\longleftrightarrow$ FBBSTs

Comparisons and Partitioning Stages

Consider quicksort on a random permutation of length n.

C_n = number of key comparison.

P_n = number of recursive calls (=partitioning stages).

Median-of-$2t+1$ Quicksort $\longleftrightarrow$ FBBSTs

Theorem (Chern, F., Hwang, Neininger (2015+))

- *For $0 \le t \le 58$, we have*

$$\rho(C_n, P_n) \to 0.$$

- *For $t \ge 59$, we have that C_n and P_n are asymptotically dependent.*

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.
- We proved similar results for variants of m-ary search trees such as fringe balanced binary search trees and quadtrees.

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.
- We proved similar results for variants of m-ary search trees such as fringe balanced binary search trees and quadtrees.
- Our result for the correlation are surprising and counterintuitive.

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.
- We proved similar results for variants of m-ary search trees such as fringe balanced binary search trees and quadtrees.
- Our result for the correlation are surprising and counterintuitive.
- Heuristic explanation of our results?

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.
- We proved similar results for variants of m-ary search trees such as fringe balanced binary search trees and quadtrees.
- Our result for the correlation are surprising and counterintuitive.
- Heuristic explanation of our results?
- Similar surprises for digital trees?

Summary and Open Questions

- We studied dependencies between shape parameters in random m-ary search trees and discovered further phase changes.
- We proved similar results for variants of m-ary search trees such as fringe balanced binary search trees and quadtrees.
- Our result for the correlation are surprising and counterintuitive.
- Heuristic explanation of our results?
- Similar surprises for digital trees?
- How about random $\sqrt{n}$-trees?