Dependencies between Shape Parameters in Random Log-Trees

Michael Fuchs
Institute of Applied Mathematics
National Chiao Tung University

Hsinchu, Taiwan

Chennai, July 10, 2015
Random Trees

- Theory
  - Combinatorial
  - Probabilistical
  - Analytical

- Applications
  - Computer Science
  - Information Theory
  - Mathematical Biology
  - Chemistry
$\sqrt{n}$-Trees vs. Log-Trees
Random Log-Trees

Trees are equipped with a random model. Average height of logarithmic order. Properties are described via Shape Parameters.
Trees are equipped with a random model

→ Random Trees
Trees are equipped with a random model

→ Random Trees

Average height of logarithmic order

→ Random Log-Trees
Trees are equipped with a random model

→ Random Trees

Average height of logarithmic order

→ 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.
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 → Random BSTs
Shape parameters become random variables
Input:
6, 2, 4, 8, 7, 1, 5, 3, 10, 9
Input:
6, 2, 4, 8, 7, 1, 5, 3, 10, 9
Input:
6, 2, 4, 8, 7, 1, 5, 3, 10, 9
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
Input:
6, 2, 4, 8, 7, 1, 5, 3, 10, 9
Input:
6, 2, 4, 8, 7, 1, 5, 3, 10, 9
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 → Random BSTs
Input:

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

If every permutation of the input sequence is equally likely

→ Random BSTs

Shape parameters become random variables
Examples of Shape Parameters

- **Height** ($= \text{maximal root-distance}$)
- **Depth** ($= \text{root-distance of a random node}$)
- **Total Path Length** ($= \text{sum of all root-distances}$)
- **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.)
Donald E. Knuth

Notes on Open Addressing
Analysis of Algorithms and Related Fields

- Complex Analysis
- Probability Theory
- Asymptotic Analysis
- Analysis of Algorithms
  - Number Theory
  - Analytic Combinatorics

Dependencies in Log-Trees

Chennai, July 10, 2015

Michael Fuchs (NCTU)
Analytic Combinatorics

Philipppe Flajolet and
Robert Sedgwick

Analytic Combinatorics
in Several
Variables

Robin Pemantle
Mark C. Wilson
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

![Diagram of m-ary search trees](image)

\( m = 3 \)

\( m = 4 \)

If permutations are equally likely \( \rightarrow \) random \( m \)-ary search trees
Size, KPL, and NPL

- **Size** (or Storage Requirement)
  
  Number of nodes holding keys. Only random if $m \geq 2$.
  
  $S_n = \text{size of a random } m\text{-ary search tree built from } n \text{ keys.}$

Size, KPL, and NPL

- **Size** (or Storage Requirement)
  Number of nodes holding keys. Only random if $m \geq 2$.
  
  \[ S_n = \text{size of a random } m\text{-ary search tree built from } n \text{ keys.} \]

- **Key Path Length** (KPL)
  Sum of all key-distances to the root.
  
  \[ K_n = \text{KPL of a random } m\text{-ary search tree built from } n \text{ keys.} \]
Size, KPL, and NPL

- **Size** (or Storage Requirement)
  Number of nodes holding keys. Only random if $m \geq 2$.
  
  $S_n = \text{size of a random } m\text{-ary search tree built from } n \text{ keys.}$

- **Key Path Length** (KPL)
  Sum of all key-distances to the root.
  
  $K_n = \text{KPL of a random } m\text{-ary search tree built from } n \text{ keys.}$

- **Node Path Length** (NPL)
  Sum of all node-distances to the root.
  
  $N_n = \text{NPL of a random } m\text{-ary search tree built from } n \text{ keys.}$
Knuth (1973):

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

where

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

and \( H_m \) are the Harmonic numbers.
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):

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

where $\alpha$ is the real part of the second largest zero of

$$\Lambda(z) = z(z + 1) \cdots (z + m - 2) - m!.$$
Mahmoud and Pittel (1989):

\[ \text{Var}(S_n) \sim \begin{cases} 
C_S n, & \text{if } m \leq 26; \\
F_1(\beta \log n)n^{2\alpha-2}, & \text{if } m \geq 27,
\end{cases} \]

where \( \lambda = \alpha + i\beta \) is the second largest zero of \( \Lambda(z) \).
Mahmoud and Pittel (1989):

\[
Var(S_n) \sim \begin{cases} 
  C_S n, & \text{if } m \leq 26; \\
  F_1(\beta \log n) n^{2\alpha - 2}, & \text{if } m \geq 27,
\end{cases}
\]

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

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

\[
F_1(z) = 2 \frac{|A|^2}{|\Gamma(\lambda)|^2} \left( -1 + \frac{m!(m-1)|\Gamma(\lambda)|^2}{\Gamma(2\alpha + m - 2) - m!\Gamma(2\alpha - 1)} \right) \\
+ 2 \Re \left( \frac{A^2 e^{2iz}}{\Gamma(\lambda)^2} \left( -1 + \frac{m!(m-1)\Gamma(\lambda)^2}{\Gamma(2\lambda + m - 2) - m!\Gamma(2\lambda - 1)} \right) \right)
\]

with \( A = 1/(\lambda(\lambda - 1) \sum_{0 \leq j \leq m-2} \frac{1}{j+\lambda}) \).
Size: Phase Change for Limit Law

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

For $3 \leq m \leq 26$,

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

where $N(0, 1)$ is the standard normal distribution.
Theorem (Mahmoud & Pittel (1989); Lew & Mahmoud (1994))

For $3 \leq m \leq 26$, 

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

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

Theorem (Chern & Hwang (2001))

For $m \geq 27$, 

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

does not converge to a fixed limit law.
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):

\[ \text{Var}(K_n) \sim C_K n^2, \]

where

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

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

\[ \text{Var}(K_n) \sim C_K n^2, \]

where

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

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

So, no phase change here for the variance!
Theorem (Neininger & Rüschendorf (1999))

We have,

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

where \( K \) is the unique solution of

\[ X \overset{d}{=} \sum_{1 \leq r \leq m} V_r X^{(r)} + 2\phi \sum_{1 \leq r \leq m} V_r \log V_r \]

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)

\[ N_n = \text{sum of all node-distances in an } m\text{-search tree built from } n \text{ keys.} \]
Node Path Length (NPL)

\[ N_n = \text{sum of all node-distances in an } m\text{-search tree built from } n \text{ keys.} \]

**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)

\[ N_n = \text{sum of all node-distances in an } m\text{-search tree built from } n \text{ keys.} \]

**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,

\[
\begin{align*}
S_n & \stackrel{d}{=} S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)} + 1, \\
N_n & \stackrel{d}{=} N_{I_1}^{(1)} + \cdots + N_{I_m}^{(m)} + S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)}.
\end{align*}
\]
**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,

\[
\begin{align*}
S_n &\overset{d}{=} S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)} + 1, \\
N_n &\overset{d}{=} N_{I_1}^{(1)} + \cdots + N_{I_m}^{(m)} + S_{I_1}^{(1)} + \cdots + S_{I_m}^{(m)}.
\end{align*}
\]

So, one expects a **strong positive dependence** between \( S_n \) and \( N_n \)!
Theorem (Chern, F., Hwang, Neininger (2015+))

We have,

\[
\text{Cov}(S_n, N_n) \sim \begin{cases} 
    C_R n \log n, & \text{if } 3 \leq m \leq 13; \\
    \phi F_2(\beta \log n)n^\alpha, & \text{if } m \geq 14,
\end{cases}
\]

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

\[
\text{Var}(N_n) \sim \phi^2 C_K n^2.
\]
Theorem (Chern, F., Hwang, Neininger (2015+))

We have,

\[
\text{Cov}(S_n, N_n) \sim \begin{cases} 
C_R n \log n, & \text{if } 3 \leq m \leq 13; \\
\phi F_2(\beta \log n)n^\alpha, & \text{if } m \geq 14,
\end{cases}
\]

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

\[
\text{Var}(N_n) \sim \phi^2 C_K n^2.
\]

Thus (!),

\[
\rho(S_n, N_n) \begin{cases} 
\rightarrow 0, & \text{if } 3 \leq m \leq 26; \\
\sim \frac{F_2(\beta \log n)}{\sqrt{C_K F_1(\beta \log n)}}, & \text{if } m \geq 27.
\end{cases}
\]
Size and NPL: Correlation (ii)

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

The graph shows the periodic function $\rho(S_n, N_n)$ for different values of $m$. The function oscillates between -1 and 1, indicating the correlation between $S_n$ and $N_n$ for each value of $m$. The graph is plotted over a range of $n$ values from 0 to 200,000.
Pearson’s Correlation Coefficient

**Pearson:** for RVs $X$ and $Y$

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

Measures linear dependence between $X$ and $Y$!
Pearson’s Correlation Coefficient

**Pearson**: for RVs $X$ and $Y$

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

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!

Michael Fuchs (NCTU)  
Dependencies in Log-Trees  
Chennai, July 10, 2015  
21 / 31
Pearson’s Correlation Coefficient

**Pearson**: for RVs $X$ and $Y$

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

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 $Y$

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

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,

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

where $N$ has a standard normal distribution.

Moreover, $N$ and $K$ are independent!
Theorem (Chern, F., Hwang, Neininger (2015+))

Consider

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

Then,

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

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,

$$\ell_2(Y_n, (\mathcal{R}(n^{\beta/\alpha} \Lambda), K)) \to 0,$$

where $\ell_2$ is the minimal $L_2$-metric and $\Lambda$ is the unique solution of

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

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

---

Michael Fuchs (NCTU)  
Dependencies in Log-Trees  
Chennai, July 10, 2015  23 / 31
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 $3, 1, 2 \Rightarrow 3, 1 \Rightarrow 3, 1, 2 \Rightarrow 2, 1, 3$

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

$T_n =$ root-distances of nodes with subtrees of size at least $2t + 1$. 
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 $3, 1, 2$

3
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 $3, 1, 2$

![Diagram showing the reorganization process with a sequence of numbers and a tree structure](image)
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 $3, 1, 2$

\[
\begin{align*}
3 & \quad \Rightarrow \quad 3 \\
1 & \quad \Rightarrow \quad 1 \\
& \quad \Rightarrow \quad 3 \\
& \quad \Rightarrow \quad 1
\end{align*}
\]
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

\[
\begin{align*}
3 & \Rightarrow \quad 1 \quad \Rightarrow \quad 1 \quad \Rightarrow \\
\ & \quad 2 \quad \quad \ & \quad 3
\end{align*}
\]

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

$T_n = \text{root-distances of nodes with subtrees of size at least } 2t + 1.$
Chern and Hwang (2001):

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

where \( \alpha_t \) 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):

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

where \( \alpha_t \) 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,

\[
\text{Var}(S_n) \sim \begin{cases} 
D_S n, & \text{if } 1 \leq t \leq 58; \\
G_1(\beta_t \log n)n^{2\alpha_t-2}, & \text{if } t \geq 59,
\end{cases}
\]

\[
\text{Cov}(S_n, T_n) \sim \begin{cases} 
D_R n, & \text{if } 1 \leq t \leq 28; \\
G_2(\beta_t \log n)n^{\alpha_t}, & \text{if } t \geq 29,
\end{cases}
\]

\[
\text{Var}(T_n) \sim D_T n^2,
\]

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 \(\Lambda_t(z)\).
Theorem (Chern, F., Hwang, Neininger (2015+))

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

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

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

$$X \xrightarrow{d} V X^{(1)} + (1 - V) X^{(2)} + D_X^{-1/2}$$

$$+ \frac{1}{D_X^{1/2}(H_{2t+2} - H_{t+1})}(V \log V + (1 - V) \log(1 - V)),$$

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

\[ \ell_2(Z_n, (\Re(n^{i\beta}\Lambda), T)) \to 0, \]

where \( \Lambda \) is the unique solution of

\[ W \overset{d}{=} V^{\lambda_t}W^{(1)} + (1 - V)^{\lambda_t}W^{(2)} \]

with \( W^{(i)} \) independent copies of \( W \).
Median-of-$2t + 1$ Quicksort


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


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


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


One of the most important algorithms 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


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


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


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 \(\rightarrow\) **Median-of-\(2t + 1\) Quicksort**
Consider quicksort on a random permutation of length $n$. 

Comparisons and Partitioning Stages

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

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

For $0 \leq t \leq 58$, we have $\rho(C_n, P_n) \rightarrow 0$.

For $t \geq 59$, we have that $C_n$ and $P_n$ are asymptotically dependent.
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).
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-2$t + 1$ Quicksort $\leftrightarrow$ FBBSTs
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-2\( t \) + 1 Quicksort \( \leftrightarrow \) FBBSTs

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

- For \( 0 \leq t \leq 58 \), we have
  \[
  \rho(C_n, P_n) \to 0.
  \]

- For \( t \geq 59 \), we have that \( C_n \) and \( P_n \) are asymptotically dependent.
We studied dependencies between shape parameters in random \( m \)-ary search trees and discovered further phase changes.

Heuristic explanation of our results?

Similar surprises for digital trees?

How about random \( \sqrt{n} \)-trees?
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.

Heuristic explanation of our results?

Similar surprises for digital trees?

How about random $\sqrt{n}$-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.
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?
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?
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?