Understanding Search Trees via Statistical Physics

Satya N. Majumdar

Laboratoire de Physique Théorique et Modèles Statistiques,CNRS,
Université Paris-Sud, France

July 9, 2007

Collaborators:

E. Ben-Naim (Los Alamos, USA)
D.S. Dean (Toulouse, FRANCE)
P.L. Krapivsky (Boston, USA)

Sorting and Search

The Goal: Store data efficiently so that the search time is minimum

Ex: A random sequence of $N = 10$ integers: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Sorting and Search

The Goal: Store data efficiently so that the search time is minimum

Ex: A random sequence of $N = 10$ integers: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Linear Sorting: Store the data sequentially onto a linear table

$$\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$$

Search for 7: Search proceeds sequentially by comparison

$$t_{\text{search}} = 10 \sim O(N) \rightarrow \text{BAD}$$

Tree Sorting: of $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

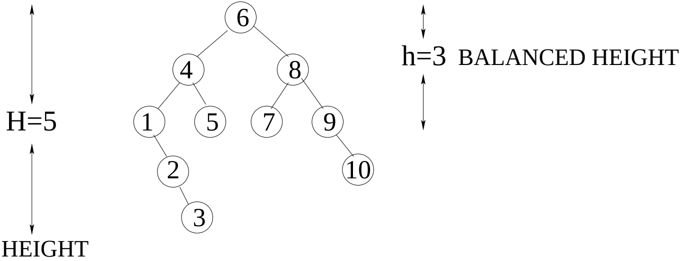


Figure: Binary Search Tree with $N = 10$ Elements.

$t_{\rm search} = {\rm Depth} = D$. Roughly $2^D \sim N$ implying: $t_{\rm search} \sim O(\log N) \rightarrow$ BETTER

Tree Sorting: of $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

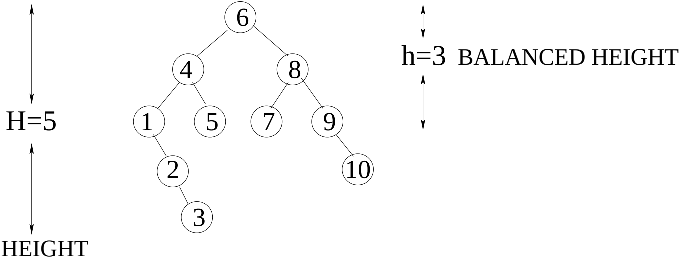


Figure: Binary Search Tree with $N = 10$ Elements.

$t_{\rm search} = {\rm Depth} = D$. Roughly $2^D \sim N$ implying: $t_{\rm search} \sim O(\log N) \rightarrow$ BETTER

• HEIGHT $H = 5$: Distance of the farthest node from the root= Maximum possible time to search an element $\rightarrow$ WORST CASE SCENARIO

Tree Sorting: of $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

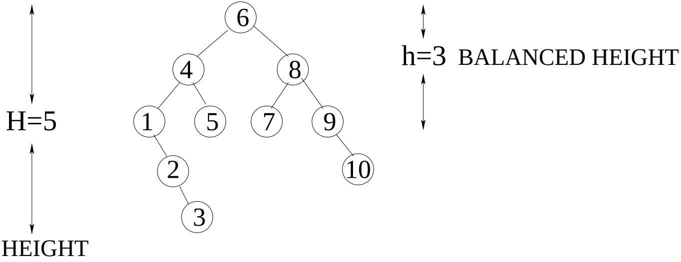


Figure: Binary Search Tree with $N = 10$ Elements.

$t_{\rm search} = {\rm Depth} = D$. Roughly $2^D \sim N$ implying: $t_{\rm search} \sim O(\log N) \to$ BETTER

• HEIGHT $H = 5$: Distance of the farthest node from the root= Maximum possible time to search an element $\to$ WORST CASE SCENARIO

• BALANCED HEIGHT $h = 3$: Depth upto which the tree is balanced

Generalization to m-ary Search Trees

$m = 2 \rightarrow$ Binary Tree

Random Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Each node can contain atmost $(m-1)$ elements.

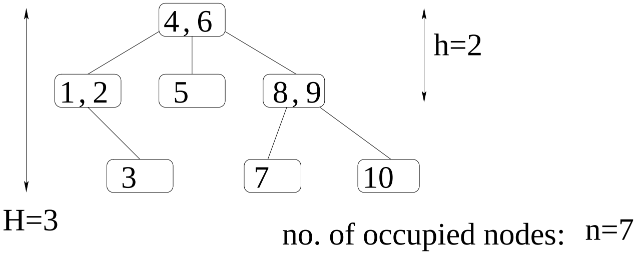


Figure: $m = 3$-ary Search Tree with $N = 10$ Elements

Generalization to m-ary Search Trees

$m = 2 \rightarrow$ Binary Tree
Random Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$
Each node can contain atmost $(m-1)$ elements.

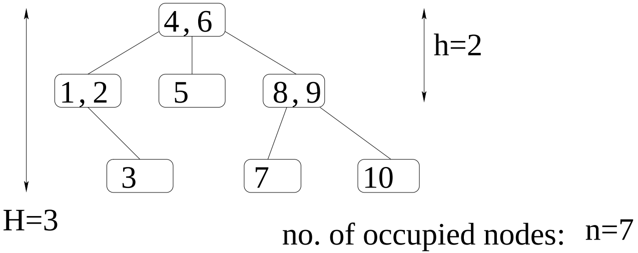


Figure: $m = 3$-ary Search Tree with $N = 10$ Elements

$H = 3 \rightarrow$ HEIGHT.

Generalization to m-ary Search Trees

$m = 2 \rightarrow$ Binary Tree
Random Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$
Each node can contain atmost $(m-1)$ elements.

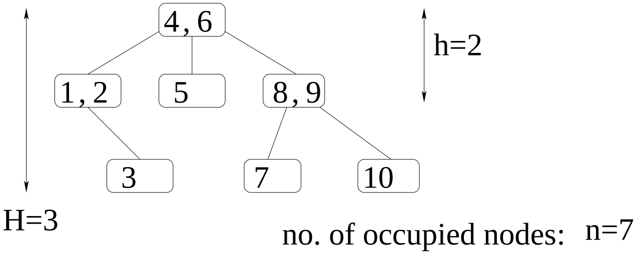


Figure: $m = 3$-ary Search Tree with $N = 10$ Elements

$H = 3 \rightarrow$ HEIGHT.

$h = 2 \rightarrow$ BALANCED HEIGHT.

Generalization to m-ary Search Trees

$m = 2 \rightarrow$ Binary Tree
Random Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$
Each node can contain atmost $(m-1)$ elements.

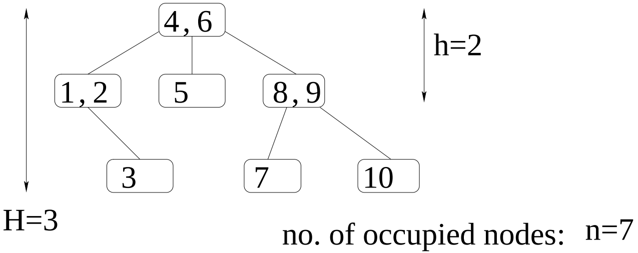


Figure: $m = 3$-ary Search Tree with $N = 10$ Elements

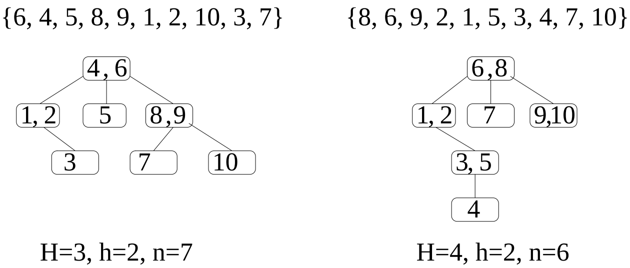
$H = 3 \rightarrow$ HEIGHT.

$h = 2 \rightarrow$ BALANCED HEIGHT.

No. of NON-EMPTY nodes: $n = 7 \rightarrow$ No. of nodes required to store the data

Random m-ary Search Tree Model: $RmST$

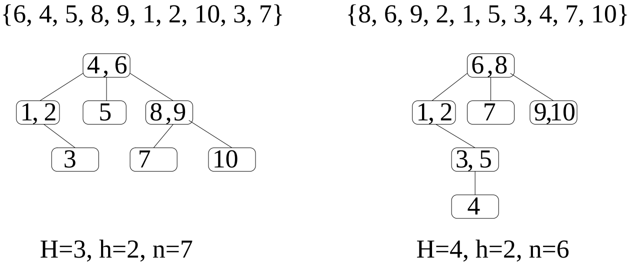
$N = 10$ data elements: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
Each permutation $\rightarrow$ an m-ary tree.



In the RmST model: All $N!$ permuations are equally likely $\rightarrow$ RANDOM DATA.

Random m-ary Search Tree Model: $RmST$

$N = 10$ data elements: $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$
Each permutation → an m-ary tree.



In the RmST model: All $N!$ permuations are equally likely → RANDOM DATA.

Q: Statistics of HEIGHT H_N, BALANCED HEIGHT h_N and the no. of NON-EMPTY NODES n_N for RANDOM data of size N?

Asymptotic Results for RmST: for large data size N

(1) Height H_N:

- $\langle H_N \rangle \approx a_m \log(N) + b_m \; \log(\log(N))$ (??) $+ \ldots$

Asymptotic Results for RmST: for large data size N

(1) Height H_N:

- $\langle H_N \rangle \approx a_m \log(N) + b_m \ \log(\log(N))$ (??) $+ \ldots$
- $\mathrm{Var}(H_N) \approx O(1)$

Asymptotic Results for RmST: for large data size N

(1) Height H_N:

- $\langle H_N \rangle \approx a_m \log(N) + b_m \ \log(\log(N))$ (??) $+ \ldots$
- $\mathrm{Var}(H_N) \approx O(1)$

(2) Balanced Height h_N: Depth upto which the tree is balanced.

- $\langle h_N \rangle \approx c_m \log(N) + d_m \ \log(\log(N))$ (??) $+ \ldots$
- $\mathrm{Var}(h_N) \approx O(1)$

Asymptotic Results for RmST: for large data size N

(1) Height H_N:

- $\langle H_N \rangle \approx a_m \log(N) + b_m \log(\log(N))$ (??) $+ \ldots$
- $\mathrm{Var}(H_N) \approx O(1)$

(2) Balanced Height h_N: Depth upto which the tree is balanced.

- $\langle h_N \rangle \approx c_m \log(N) + d_m \log(\log(N))$ (??) $+ \ldots$
- $\mathrm{Var}(h_N) \approx O(1)$

Binary Tree ($m = 2$): $a_2 = 4.31107\ldots$ and $c_2 = 0.3733\ldots$ (Devroye, 87).

Asymptotic Results for RmST: for large data size N

(1) Height H_N:

- $\langle H_N \rangle \approx a_m \log(N) + b_m \ \log(\log(N))$ (??) $+\ldots$
- $\mathrm{Var}(H_N) \approx O(1)$

(2) Balanced Height h_N: Depth upto which the tree is balanced.

- $\langle h_N \rangle \approx c_m \log(N) + d_m \ \log(\log(N))$ (??) $+\ldots$
- $\mathrm{Var}(h_N) \approx O(1)$

Binary Tree ($m = 2$): $a_2 = 4.31107\ldots$ and $c_2 = 0.3733\ldots$ (Devroye, 87).

The correction terms $\rightarrow$ conjectured by Hattori and Ochiai (simulations, 2001).

Other results by Robson (2001-), Reed (2001-), Drmota (2001-), Flajolet, ...

Asymptotic Results for RmST: for large data size N...continued

(3) No. of NON-EMPTY Nodes n_N: No. of nodes required to store the data of size N.

$$\langle n_N \rangle \approx \alpha_m N + \ldots .$$

Asymptotic Results for RmST: for large data size N...continued

(3) No. of NON-EMPTY Nodes n_N: No. of nodes required to store the data of size N.

$$\langle n_N \rangle \approx \alpha_m N + \ldots$$

A striking PHASE TRANSITION occurs for the Variance: $\nu_N = \langle \left(n_N - \langle n_N \rangle\right)^2 \rangle$.

$$\begin{aligned} \nu_N &\sim N && \text{for } m \leq 26 \\ &\sim N^{2\theta(m)} && \text{for } m > 26 \end{aligned}$$

(Chern & Hwang, 2001).

Asymptotic Results for RmST: for large data size N...continued

(3) No. of NON-EMPTY Nodes n_N: No. of nodes required to store the data of size N.

$$\langle n_N \rangle \approx \alpha_m N + \ldots .$$

A striking PHASE TRANSITION occurs for the Variance: $\nu_N = \langle \left(n_N - \langle n_N \rangle\right)^2 \rangle$.

$$\nu_N \sim N \qquad \text{for } m \leq 26$$
$$\sim N^{2\theta(m)} \qquad \text{for } m > 26$$ (Chern & Hwang, 2001).

Q: Why 26? What is the mechanism of this Phase Transition and how generic is it? Can one calculate $\theta(m)$ exactly ?

Our Results:

- Mapping to a FRAGMENTATION Process $\longrightarrow$ Dynamical Process

Our Results:

• Mapping to a FRAGMENTATION Process $\rightarrow$ Dynamical Process

• Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

$\rightarrow$ A number of asymptotically EXACT results.

Our Results:

• Mapping to a FRAGMENTATION Process → Dynamical Process

• Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

→ A number of asymptotically EXACT results.

Ex: we calculate the constants a_m, b_m, c_m, d_m for all m as roots of transcendental equations.

Our Results:

• Mapping to a FRAGMENTATION Process → Dynamical Process

• Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

→ A number of asymptotically EXACT results.

Ex: we calculate the constants a_m, b_m, c_m, d_m for all m as roots of transcendental equations.

• Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2 - 1)]$.

Our Results:

• Mapping to a FRAGMENTATION Process → Dynamical Process

• Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

→ A number of asymptotically EXACT results.

Ex: we calculate the constants a_m, b_m, c_m, d_m for all m as roots of transcendental equations.

• Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2 - 1)]$.

• We show that $m_c = 26.0461...$:

Our Results:

- Mapping to a FRAGMENTATION Process → Dynamical Process

- Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

→ A number of asymptotically EXACT results.

Ex: we calculate the constants $a_m,\ b_m,\ c_m,\ d_m$ for all m as roots of transcendental equations.

- Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2-1)]$.

- We show that $m_c = 26.0461...$:

Find $\lambda(m)$ from: $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$.

Our Results:

- Mapping to a FRAGMENTATION Process $\rightarrow$ Dynamical Process

- Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

$\rightarrow$ A number of asymptotically EXACT results.

Ex: we calculate the constants a_m, b_m, c_m, d_m for all m as roots of transcendental equations.

- Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2-1)]$.

- We show that $m_c = 26.0461\ldots$:

Find $\lambda(m)$ from: $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$.

The critical value m_c is obtained by setting, $Re[\lambda(m) = 1/2]$.

Our Results:

- Mapping to a FRAGMENTATION Process $\rightarrow$ Dynamical Process

- Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

$\rightarrow$ A number of asymptotically EXACT results.

Ex: we calculate the constants $a_m,\ b_m,\ c_m,\ d_m$ for all m as roots of transcendental equations.

- Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2-1)]$.

- We show that $m_c = 26.0461...$:

Find $\lambda(m)$ from: $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$.

The critical value m_c is obtained by setting, $Re[\lambda(m) = 1/2]$.

- For $m > m_c = 26.0461...$, $\theta(m) = \lambda(m)$. (D.S. Dean and S.M., 2002).

Our Results:

- Mapping to a FRAGMENTATION Process → Dynamical Process

- Analysis of the FRAGMENTATION process using a variety of statistical physics techniques such as the Travelling Front method (for HEIGHTS and BALANCED HEIGHTS) and a Backward Fokker-Planck approach (for the no. of NON-EMPTY Nodes).

→ A number of asymptotically EXACT results.

Ex: we calculate the constants a_m, b_m, c_m, d_m for all m as roots of transcendental equations.

- Scaling Relation between a_m and b_m: $b_2 = -3a_2/[2(a_2-1)]$.

- We show that $m_c = 26.0461...$:

Find $\lambda(m)$ from: $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$.

The critical value m_c is obtained by setting, $Re[\lambda(m) = 1/2]$.

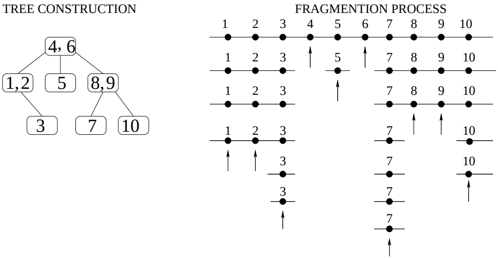
- For $m > m_c = 26.0461...$, $\theta(m) = \lambda(m)$. (D.S. Dean and S.M., 2002).

- Various other generalizations: Vector Data

The Mapping to a Fragmentation Process

Construction of the Tree → Dynamical Fragmention Process: Split an interval into m pieces with the break points chosen randomly. An interval can split iff it contains atleast one point.

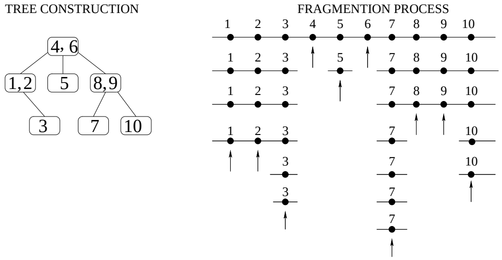
Ex: Consider the data: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$ on a $(m = 3)$-ary tree



The Mapping to a Fragmentation Process

Construction of the Tree → Dynamical Fragmention Process: Split an interval into m pieces with the break points chosen randomly. An interval can split iff it contains atleast one point.

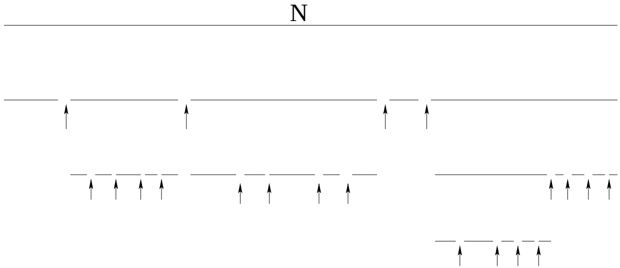
Ex: Consider the data: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$ on a ($m = 3$)-ary tree



NOTE:

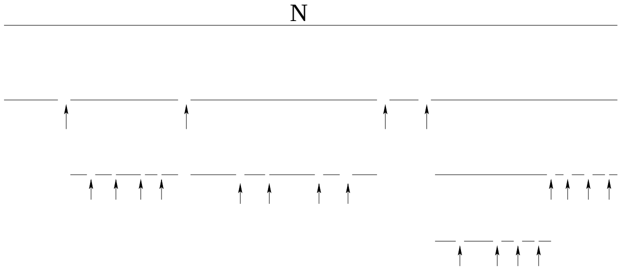
No. of NONEMPTY nodes n=7= No. of SPLITTING EVENTS

Fragmentation Process: Continuum Limit



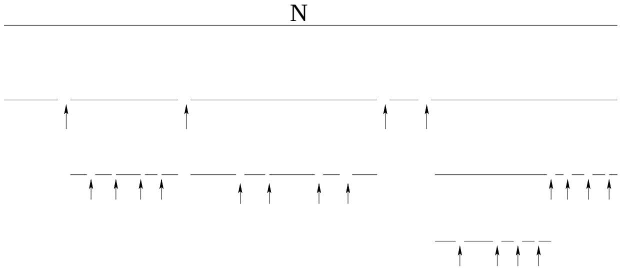
1. Start with a stick of length N.

Fragmentation Process: Continuum Limit



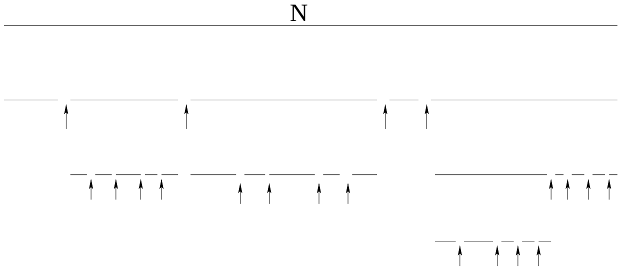
1. Start with a stick of length N.
2. Choose $(m-1)$ break points randomly and split the stick into m pieces.

Fragmentation Process: Continuum Limit



1. Start with a stick of length N.
2. Choose $(m-1)$ break points randomly and split the stick into m pieces.
3. Examine each piece and if its length $> N_0 = 1 =$ Threshold, again split it randomly into further m pieces. Stop splitting if length $< 1 =$.

Fragmentation Process: Continuum Limit



1. Start with a stick of length N.
2. Choose $(m-1)$ break points randomly and split the stick into m pieces.
3. Examine each piece and if its length $> N_0 = 1 =$ Threshold, again split it randomly into further m pieces. Stop splitting if length $< 1 =$.
4. Repeat the process till all pieces have length < 1 and then STOP.

DICTIONARY Between the Search Tree and the Fragmentation Process:

m-ary SEARCH TREE $\equiv$ FRAGMENTATION PROCESS

DICTIONARY Between the Search Tree and the Fragmentation Process:

m-ary SEARCH TREE $\equiv$ FRAGMENTATION PROCESS

Height H_N:

- Prob[$H_N < n$]= Prob[$l_1 < 1$, $l_2 < 1$, ... after n steps]

Balanced Height h_N:

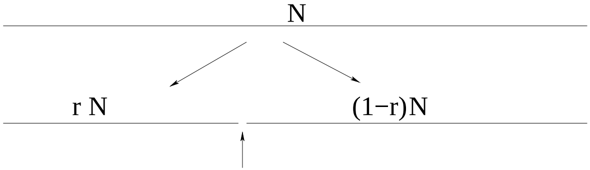
- Prob[$h_N > n$]=Prob[$l_1 > 1$, $l_2 > 1$, ... after n steps]

Number of Nonempty Nodes n_N ($m > 2$):

- Prob[$n_N = n$]= Prob[there are n SPILLITING EVENTS till the end of the Fragmentation process].

Analysis of HEIGHT H_N

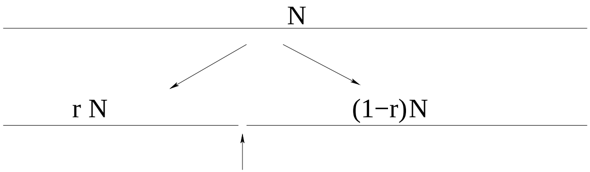
P(n,N)= Prob[$H_N < n$]= Prob[$l_1 < 1,\ l_2 < 1,\ \ldots$ after n steps starting with initial length N]



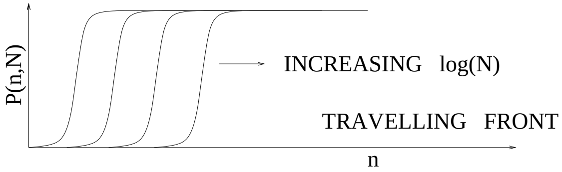
Recursion: $P(n,N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1,(1-r)N)\, dr$ starting with $P(n,1) = \theta(n-1)$.

Analysis of HEIGHT H_N

P(n,N)= Prob[$H_N < n$]= Prob[$l_1 < 1,\ l_2 < 1,\ \ldots$ after n steps starting with initial length N]



Recursion: $P(n,N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1,(1-r)N)\, dr$ starting with $P(n,1) = \theta(n-1)$.



Travelling Front in Fisher Equation

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

Travelling Front in Fisher Equation

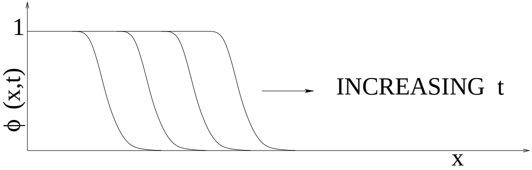
$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

$\phi(x) = 1 \rightarrow$ STABLE Fixed point. $\phi(x) = 0 \rightarrow$ UNSTABLE Fixed point.

Travelling Front in Fisher Equation

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

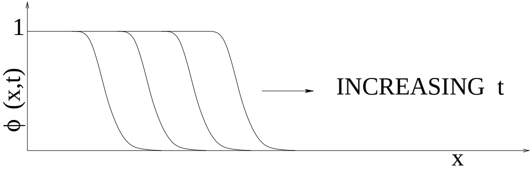
$\phi(x) = 1 \rightarrow$ STABLE Fixed point. $\phi(x) = 0 \rightarrow$ UNSTABLE Fixed point.



Travelling Front in Fisher Equation

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

$\phi(x) = 1 \rightarrow$ STABLE Fixed point. $\phi(x) = 0 \rightarrow$ UNSTABLE Fixed point.



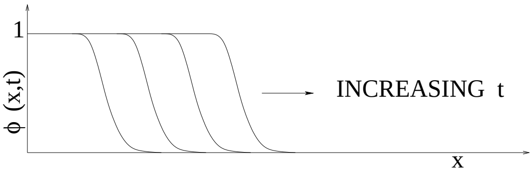
Travelling Front: $\phi(x,t) = f\,(x - x_f(t))$ for large t, where the front position

$$x_f(t) \sim \mathrm{v}\; t + \ldots.$$

Travelling Front in Fisher Equation

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

$\phi(x) = 1 \rightarrow$ STABLE Fixed point. $\phi(x) = 0 \rightarrow$ UNSTABLE Fixed point.



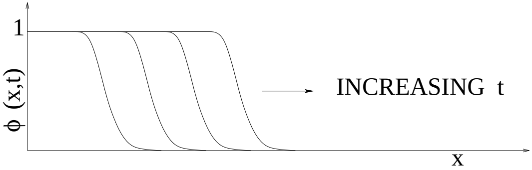
Travelling Front: $\phi(x,t) = f\,(x - x_f(t))$ for large t, where the front position

$$x_f(t) \sim \mathrm{v}\ t + \ldots.$$

Q: How to determine the Front Velocity v?

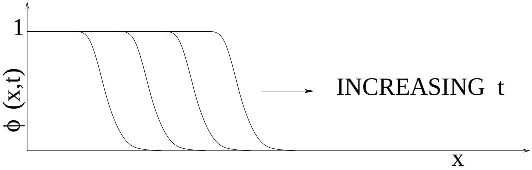
Kolmogorov's Velocity Selection Principle:

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.



Kolmogorov's Velocity Selection Principle:

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.

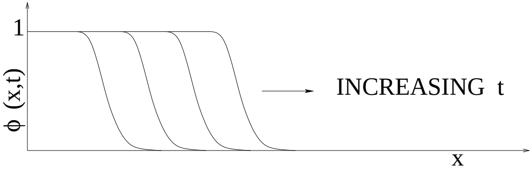


Linearize near the tail $\rightarrow$ $\phi(x,t) \sim \exp[-\lambda(x - vt)]$

DISPERSION RELATION: $v(\lambda) = \lambda + \frac{1}{\lambda}$ $\rightarrow$ minimum at $\lambda^* = 1$.

Kolmogorov's Velocity Selection Principle:

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.



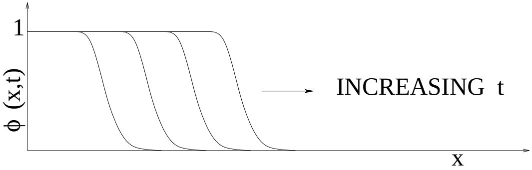
Linearize near the tail $\rightarrow$ $\phi(x,t) \sim \exp[-\lambda(x - vt)]$

DISPERSION RELATION: $v(\lambda) = \lambda + \frac{1}{\lambda}$ $\rightarrow$ minimum at $\lambda^* = 1$.

For sharp initial condition, Front velocity $v = v(\lambda^*) = 2$.

Kolmogorov's Velocity Selection Principle:

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.



Linearize near the tail $\rightarrow$ $\phi(x,t) \sim \exp[-\lambda(x - vt)]$

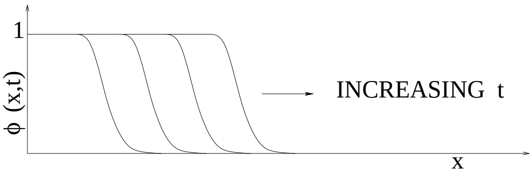
DISPERSION RELATION: $v(\lambda) = \lambda + \frac{1}{\lambda}$ $\rightarrow$ minimum at $\lambda^* = 1$.

For sharp initial condition, Front velocity $v = v(\lambda^*) = 2$.

More generally, $\phi(x,t) \sim \exp[-\lambda(x - x_f(t))]$

Kolmogorov's Velocity Selection Principle:

$\partial_t \phi(x,t) = \partial_x^2 \phi(x,t) + \phi - \phi^2$.



Linearize near the tail $\rightarrow \phi(x,t) \sim \exp[-\lambda(x - vt)]$

DISPERSION RELATION: $v(\lambda) = \lambda + \frac{1}{\lambda}$ $\rightarrow$ minimum at $\lambda^* = 1$.

For sharp initial condition, Front velocity $v = v(\lambda^*) = 2$.

More generally, $\phi(x,t) \sim \exp[-\lambda(x - x_f(t))]$

$x_f(t) \approx v(\lambda^*)t - \frac{3}{2\lambda^*}\log t + \ldots$

(Bramson, Brunet & Derrida, van Saarloos, ...)

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n, N) = \int_0^1 P\left(n-1, rN\right) P\left(n-1, (1-r)N\right) dr$

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n,N) = \int_0^1 P\left(n-1, rN\right) P\left(n-1, (1-r)N\right) dr$

$P(n,N) = Prob[H_N < n] \approx f[n - n_f(N)]$ asymptotically. $t \equiv \log N \longrightarrow$ correct variable.

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n,N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1,(1-r)N)\, dr$

$P(n,N) = Prob[H_N < n] \approx f[n - n_f(N)]$ asymptotically. $t \equiv \log N \longrightarrow$ correct variable.

Linearize near the tail: $P(n,N) \approx 1 - \exp\left[-\lambda\,(n - v(\lambda)\log N)\right]$

$\longrightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n,N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1,(1-r)N)\, dr$

$P(n,N) = Prob[H_N < n] \approx f[n - n_f(N)]$ asymptotically. $t \equiv \log N \longrightarrow$ correct variable.

Linearize near the tail: $P(n,N) \approx 1 - \exp\left[-\lambda\,(n - v(\lambda)\log N)\right]$

$\longrightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Minimize $v(\lambda) \longrightarrow \lambda^* = 0.76804\ldots$.

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\rightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n,N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1, (1-r)N)\, dr$

$P(n,N) = Prob[H_N < n] \approx f\,[n - n_f(N)]$ asymptotically. $t \equiv \log N \rightarrow$ correct variable.

Linearize near the tail: $P(n,N) \approx 1 - \exp\left[-\lambda\,(n - v(\lambda)\log N)\right]$

$\rightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Minimize $v(\lambda) \rightarrow \lambda^* = 0.76804\ldots$.

$$\langle H_N \rangle \approx n_f(N) \approx \ v(\lambda^*)\ \log(N) \text{ - } \frac{3}{2\lambda^*}\ \log\left(\log(N)\right) + \ldots$$

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n, N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1, (1-r)N)\, dr$

$P(n, N) = Prob[H_N < n] \approx f\,[n - n_f(N)]$ asymptotically. $t \equiv \log N \longrightarrow$ correct variable.

Linearize near the tail: $P(n, N) \approx 1 - \exp\left[-\lambda\,(n - v(\lambda) \log N)\right]$

$\longrightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Minimize $v(\lambda) \longrightarrow \lambda^* = 0.76804\ldots$.

$$\langle H_N \rangle \approx n_f(N) \approx \; v(\lambda^*)\; \log(N) - \frac{3}{2\lambda^*}\; \log\left(\log(N)\right) + \ldots$$

$\longrightarrow \qquad a_2 = v(\lambda^*) = 4.31107\ldots$ and $b_2 = -\frac{3}{2\lambda^*} = -1.95303\ldots$

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\longrightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n, N) = \int_0^1 P\,(n-1, rN)\, P\,(n-1, (1-r)N)\, dr$

$P(n, N) = Prob[H_N < n] \approx f[n - n_f(N)]$ asymptotically. $t \equiv \log N \longrightarrow$ correct variable.

Linearize near the tail: $P(n, N) \approx 1 - \exp\left[-\lambda\,(n - v(\lambda)\log N)\right]$

$\longrightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Minimize $v(\lambda) \longrightarrow \lambda^* = 0.76804\ldots$.

$$\langle H_N \rangle \approx n_f(N) \approx \; v(\lambda^*)\; \log(N) \text{ - } \frac{3}{2\lambda^*}\; \log(\log(N)) + \ldots$$

$\longrightarrow \qquad a_2 = v(\lambda^*) = 4.31107\ldots$ and $b_2 = -\frac{3}{2\lambda^*} = -1.95303\ldots$

Similarly one gets a_m and b_m for all m.

Travelling Front Solution to Search Tree Height:

Kolmogorov principle $\rightarrow$ more general (not just for Fisher equation).

Applying the same strategy to the Nonlinear Tree Recursion Relation:

$P(n, N) = \int_0^1 P(n-1, rN)\, P(n-1, (1-r)N)\, dr$

$P(n, N) = Prob[H_N < n] \approx f[n - n_f(N)]$ asymptotically. $t \equiv \log N \rightarrow$ correct variable.

Linearize near the tail: $P(n, N) \approx 1 - \exp\left[-\lambda\left(n - v(\lambda)\log N\right)\right]$

$\rightarrow$ DISPERSION RELATION: $v(\lambda) = \frac{2e^{\lambda}-1}{\lambda}$ for $m = 2$.

Minimize $v(\lambda) \rightarrow \lambda^* = 0.76804\ldots$.

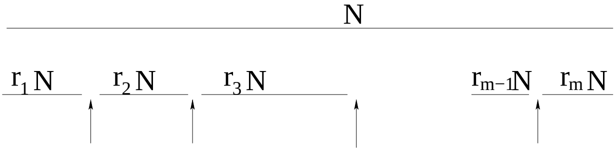
$$\langle H_N \rangle \approx n_f(N) \approx v(\lambda^*)\ \log(N) - \frac{3}{2\lambda^*}\ \log\left(\log(N)\right) + \ldots$$

$\rightarrow \quad a_2 = v(\lambda^*) = 4.31107\ldots$ and $b_2 = -\frac{3}{2\lambda^*} = -1.95303\ldots$

Similarly one gets a_m and b_m for all m.

Same strategy holds for the Balanced Height h_N.

No of Non-Empty Nodes:



$$r_1 + r_2 + r_3 + \ldots\ldots + r_m = 1$$

No. of Non-empty nodes $n(N)$ in the tree $\equiv$ Total no. of Splitting Events in the fragmentation process till the end, starting with the initial length N

Recursion:

$$n(N) \equiv n(r_1 N) + n(r_2 N) + n(r_3 N) + \cdots + n(r_m N) + 1; \qquad \sum_i^n r_i = 1$$

The marginal distribution of any fragment: $\eta_m(r) = (m-1)(1-r)^{m-2}$

Integral Equations for Average and Variance:

Average: $\mu(N) = \langle n(N) \rangle$ satisfies an integral equation:

Integral Equations for Average and Variance:

Average: $\mu(N) = \langle n(N) \rangle$ satisfies an integral equation:

$$\mu(N) = m \int_{1/N}^{1} \mu(rN)\eta_m(r)dr + 1$$

where $\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Integral Equations for Average and Variance:

Average: $\mu(N) = \langle n(N) \rangle$ satisfies an integral equation:

$$\mu(N) = m \int_{1/N}^{1} \mu(rN)\eta_m(r)dr + 1$$

where $\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Variance: $\nu(N) = \langle (n(N) - \mu(N))^2 \rangle$ satisfies another integral equation:

$$\nu(N) = m \int_{1/N}^{1} \nu(rN)\eta(r)dr + \langle (S - \langle S \rangle)^2 \rangle$$

where the Source Function $S = \sum_{i=1}^{m} \mu(r_i N)$.

These integral equations can be solved analytically:

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^{\lambda} \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^{\lambda} \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Variance: $\nu(N) = \beta_1 N + \beta_2 N^{2\lambda_2} + \beta_3 N^{2\lambda_2^*} + \beta_3 N^{\lambda_2 + \lambda_2^*} + \ldots$

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^\lambda \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Variance: $\nu(N) = \beta_1 N + \beta_2 N^{2\lambda_2} + \beta_3 N^{2\lambda_2^*} + \beta_3 N^{\lambda_2+\lambda_2^*} + \ldots$

As one tunes m, the dominant term is either N (for $m < m_c$) or $N^{2(Re\lambda_2)}$ (for $m > m_c$):

for large N,

$$\begin{aligned} \nu(N) &\sim N && \text{for } m \leq m_c \\ &\sim N^{2\theta(m)} && \text{for } m > m_c. \end{aligned}$$

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^{\lambda} \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Variance: $\nu(N) = \beta_1 N + \beta_2 N^{2\lambda_2} + \beta_3 N^{2\lambda_2^*} + \beta_3 N^{\lambda_2+\lambda_2^*} + \ldots$

As one tunes m, the dominant term is either N (for $m < m_c$) or $N^{2(Re\lambda_2)}$ (for $m > m_c$):

for large N,

$$\nu(N) \sim N \qquad \text{for } m \leq m_c$$
$$\sim N^{2\theta(m)} \quad \text{for } m > m_c.$$

m_c is determined as: Find $\lambda_2(m)$ from the root (closest to 1) of $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$. The critical m_c is obtained by setting,

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^\lambda \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \to$ marginal distribution of the fraction r.

Variance: $\nu(N) = \beta_1 N + \beta_2 N^{2\lambda_2} + \beta_3 N^{2\lambda_2^*} + \beta_3 N^{\lambda_2+\lambda_2^*} + \ldots$

As one tunes m, the dominant term is either N (for $m < m_c$) or $N^{2(Re\lambda_2)}$ (for $m > m_c$):

for large N,

$$\begin{aligned} \nu(N) &\sim N \qquad \text{for } m \leq m_c \\ &\sim N^{2\theta(m)} \quad \text{for } m > m_c. \end{aligned}$$

m_c is determined as: Find $\lambda_2(m)$ from the root (closest to 1) of $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$. The critical m_c is obtained by setting,

$$Re[\lambda_2(m) = 1/2] \qquad .$$

Mechanism of Phase Transition: Eigenvalue Crossing

For large N:

Mean: $\mu(N) \sim \alpha_0 + \alpha_1 N + \sum_{k=2}^{\infty} \alpha_K N^{\lambda_k}$

where λ_k's are zeros of: $m \int_0^1 r^{\lambda} \eta_m(r) dr = 1$ with

$\eta_m(r) = (m-1)(1-r)^{m-2} \rightarrow$ marginal distribution of the fraction r.

Variance: $\nu(N) = \beta_1 N + \beta_2 N^{2\lambda_2} + \beta_3 N^{2\lambda_2^*} + \beta_3 N^{\lambda_2 + \lambda_2^*} + \ldots$

As one tunes m, the dominant term is either N (for $m < m_c$) or $N^{2(Re\lambda_2)}$ (for $m > m_c$):

for large N,

$$\begin{aligned} \nu(N) &\sim N \quad \text{for } m \leq m_c \\ &\sim N^{2\theta(m)} \quad \text{for } m > m_c. \end{aligned}$$

m_c is determined as: Find $\lambda_2(m)$ from the root (closest to 1) of $m(m-1)\mathrm{B}(\lambda+1, m-1) = 1$. The critical m_c is obtained by setting,

$$Re[\lambda_2(m) = 1/2]$$

.

For $m > m_c = 26.0461...$, $\theta(m) = \lambda_2(m)$. (Dean and S.M., 2002).

Generalization to Vector Data

Vector Data: Quadtree Structure (Finkel and Bentley, Flajolet and Richmond)

Scalar Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Generalization to Vector Data

Vector Data: Quadtree Structure (Finkel and Bentley, Flajolet and Richmond)

Scalar Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Vector Sequence: $\{(6, 4), (4, 3), (5, 2), (8, 7) \ldots\} \rightarrow D = 2$ vector.

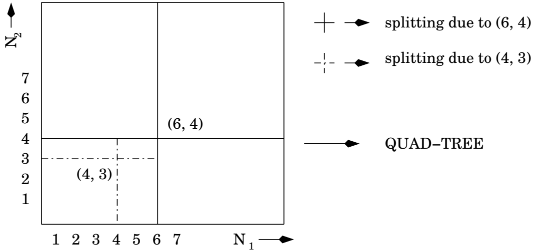
Generalization to Vector Data

Vector Data: Quadtree Structure (Finkel and Bentley, Flajolet and Richmond)

Scalar Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Vector Sequence: $\{(6, 4), (4, 3), (5, 2), (8, 7) \ldots\} \rightarrow D = 2$ vector.

Mapping to the Fragmentation Process: Break a hypercube into 2^D parts.



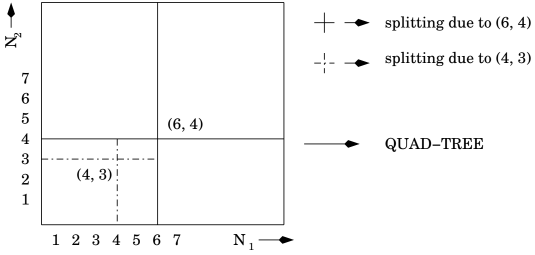
Generalization to Vector Data

Vector Data: Quadtree Structure (Finkel and Bentley, Flajolet and Richmond)

Scalar Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Vector Sequence: $\{(6, 4), (4, 3), (5, 2), (8, 7) \ldots\} \rightarrow D = 2$ vector.

Mapping to the Fragmentation Process: Break a hypercube into 2^D parts.



Q: What are the statistics of Height H_N, Balanced Height h_N and the no. of Non-empty nodes n_N for a given vector data of N D-tuples?

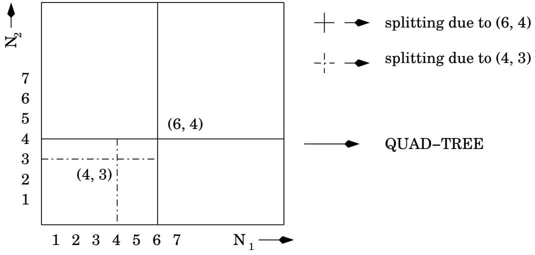
Generalization to Vector Data

Vector Data: Quadtree Structure (Finkel and Bentley, Flajolet and Richmond)

Scalar Sequence: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

Vector Sequence: $\{(6, 4), (4, 3), (5, 2), (8, 7) \ldots\} \rightarrow D = 2$ vector.

Mapping to the Fragmentation Process: Break a hypercube into 2^D parts.



Q: What are the statistics of Height H_N, Balanced Height h_N and the no. of Non-empty nodes n_N for a given vector data of N D-tuples?

Is there a PHASE TRANSITION in the variance of n_N?

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

• $\langle H_N \rangle \approx 4.31107\ldots \log(N)$ - $\frac{1.95303\ldots}{D}$ $\log(D \log(N)) + \ldots$

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N) - \frac{1.95303\ldots}{D} \log\left(D \log(N)\right) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log\left(D \log(N)\right) + \ldots$

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots\ \log(N) - \frac{1.95303\ldots}{D} \log\left(D \log(N)\right) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log\left(D \log(N)\right) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N) - \frac{1.95303\ldots}{D} \log(D \log(N)) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log(D \log(N)) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Variance ν_N has a Phase Transition:

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N) - \frac{1.95303\ldots}{D} \log(D\log(N)) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log(D\log(N)) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Variance ν_N has a Phase Transition:

$$\nu_N \sim V \qquad \text{for } D \leq D_c$$

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N) - \frac{1.95303\ldots}{D} \log(D\log(N)) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log(D\log(N)) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Variance ν_N has a Phase Transition:

$$\begin{aligned} \nu_N &\sim V && \text{for } D \leq D_c \\ &\sim V^{2\theta(D)} && \text{for } D > D_c \end{aligned}$$

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N)$ - $\frac{1.95303\ldots}{D} \log(D\log(N)) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log(D\log(N)) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Variance ν_N has a Phase Transition:

$$\begin{aligned} \nu_N &\sim V \quad &\text{for } D \leq D_c \\ &\sim V^{2\theta(D)} \quad &\text{for } D > D_c \end{aligned}$$

$$D_c = \frac{\pi}{\arcsin\left(\frac{1}{\sqrt{8}}\right)} = 8.69362\ldots$$

Exact Results for Vector Data of N D-tuples for Large N:

Height H_N :

- $\langle H_N \rangle \approx 4.31107\ldots \log(N)$ - $\frac{1.95303\ldots}{D}$ $\log(D\log(N)) + \ldots$

Balanced Height h_N :

- $\langle h_N \rangle \approx 0.37336\ldots \log(N) + \frac{0.89374\ldots}{D} \log(D\log(N)) + \ldots$

No. of Non-empty Nodes n_N : $\langle n_N \rangle \approx \frac{2}{D} V$ where $V = N^D$.

Variance ν_N has a Phase Transition:

$$\nu_N \sim V \quad \text{for } D \leq D_c$$
$$\sim V^{2\theta(D)} \quad \text{for } D > D_c$$

$$D_c = \frac{\pi}{\arcsin\left(\frac{1}{\sqrt{8}}\right)} = 8.69362\ldots$$

$\theta(D) = 2\cos(\frac{2\pi}{D}) - 1 \rightarrow$ increases continuously with D for $D > D_c$

Probability Distribution of the no. of Non-Empty Nodes n_N:

$P[n_N] \rightarrow$ GAUSSIAN for $D < D_c = 8.69362\ldots$

Probability Distribution of the no. of Non-Empty Nodes n_N:

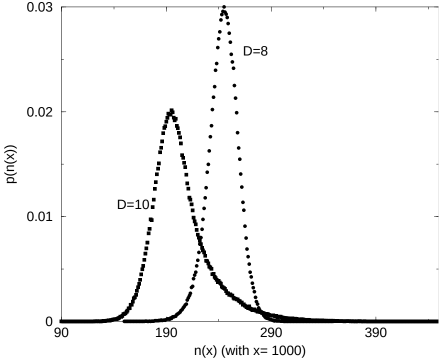
$P[n_N] \to$ GAUSSIAN for $D < D_c = 8.69362\ldots$

$P[n_V] \to$ NON-GAUSSIAN for $D > D_c = 8.69362\ldots$

Probability Distribution of the no. of Non-Empty Nodes n_N:

$P[n_N] \rightarrow$ GAUSSIAN for $D < D_c = 8.69362\ldots$

$P[n_V] \rightarrow$ NON-GAUSSIAN for $D > D_c = 8.69362\ldots$



Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics $\longrightarrow$ Exact asymptotic results.

Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics $\longrightarrow$ Exact asymptotic results.

• Going beyond Random m-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics $\longrightarrow$ Exact asymptotic results.

• Going beyond Random m-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

• Application of the Travelling Front techniques to computer science problems.

Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics $\longrightarrow$ Exact asymptotic results.

• Going beyond Random m-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

• Application of the Travelling Front techniques to computer science problems.

• A simple mechanism for the peculiar Phase Transition in the fluctuation of the number of non-empty nodes

Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics → Exact asymptotic results.

• Going beyond Random m-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

• Application of the Travelling Front techniques to computer science problems.

• A simple mechanism for the peculiar Phase Transition in the fluctuation of the number of non-empty nodes

→ A rather Generic phase transition → New Exact Results for Vector Data.

Summary and Conclusion:

• Analysis of *m*-ary search trees via techniques of statistical physics → Exact asymptotic results.

• Going beyond Random *m*-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

• Application of the Travelling Front techniques to computer science problems.

• A simple mechanism for the peculiar Phase Transition in the fluctuation of the number of non-empty nodes

→ A rather Generic phase transition → New Exact Results for Vector Data.

The same mechanism is also responsible for the phase transition in a Growing Tree Model of Aldous & Shields (1988)...Explicit Results (Dean and S.M, 2006).

Summary and Conclusion:

• Analysis of m-ary search trees via techniques of statistical physics $\longrightarrow$ Exact asymptotic results.

• Going beyond Random m-ary search trees...Digital Search Trees.. interesting connections to Diffusion Limited Aggregation (DLA) on the Bethe lattice and also to the Lempel-Ziv Data Compression Algorithm (S.M., 2003).

• Application of the Travelling Front techniques to computer science problems.

• A simple mechanism for the peculiar Phase Transition in the fluctuation of the number of non-empty nodes

$\longrightarrow$ A rather Generic phase transition $\longrightarrow$ New Exact Results for Vector Data.

The same mechanism is also responsible for the phase transition in a Growing Tree Model of Aldous & Shields (1988)...Explicit Results (Dean and S.M, 2006).

Perspectives: Lots of beautiful open problems in Sorting and Search that may be possible to handle by using statistical physics techniques.

References:

Collaborators: E. Ben-Naim, D.S. Dean and P.L. Krapivsky

- PRL, 85, 5492 (2000)
- PRE, 62, 7735 (2000)
- PRE, 63, 045101 (R) (2001)
- PRE, 64, 046121 (2001)
- PRE, 64, 035101 (R) (2001)
- PRE, 65, 036127 (2002)
- J-Phys A: Math-Gen, 35, L501 (2002)
- PRE, 68, 026103 (2003)
- J. Stat. Phys., 124, 1351 (2006)

• For a short Review see: S.M., D.S. Dean and P.L. Krapivsky, Procedings of the STATPHYS-22 (Bangalore, 2004), arXiv:cond-mat/0410498.