Understanding Search Trees Via Statistical Physics

Satya N. Majumdar

CNRS, Orsay and Toulouse, FRANCE

Collaborators: E. Ben-Naim (Los Alamos, USA), D.S. Dean (Toulouse, FRANCE) and 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: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

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\}

![Binary Search Tree](image)

Figure 1: Binary Search Tree with \(N = 10\) Elements.

- \(t_{\text{search}} = \text{Depth} = D\). Roughly \(2^D \sim N\) implying: \(t_{\text{search}} \sim O(\log N) \rightarrow \text{BETTER}\)
- \(\text{HEIGHT } H = 5\): Distance of the farthest node from the root= Maximum possible time to search an element \(\rightarrow \text{WORST CASE SCENARIO}\)
- \(\text{BALANCED HEIGHT } h = 3\): Depth upto which the tree is balanced
Generalization to \( m \)-ary Search Trees

\( n = 2 \) → Binary Tree

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

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

\[ \text{Figure 2: } m = 3\text{-ary Search Tree with } N = 10 \text{ Elements} \]

\( H = 3 \) is the \textbf{HEIGHT}. \( h = 2 \) is the \textbf{BALANCED HEIGHT}.

No. of \textbf{NON-EMPTY} nodes: \( n = 7 \) → 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.

$$
\begin{align*}
\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\} & & \{8, 6, 9, 2, 1, 5, 3, 4, 7, 10\} \\
\begin{array}{c}
4, 6 \\
1, 2 & 5 & 8, 9 \\
& 3 & 7 & 10
\end{array}
& & \\
\begin{array}{c}
6, 8 \\
1, 2 & 7 & 9, 10 \\
& 3, 5 \\
& & 4
\end{array}
\end{align*}
$$

H=3, h=2, n=7 \\
H=4, h=2, n=6

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

Q: Statistics of $\text{HEIGHT } H_N$, $\text{BALANCED HEIGHT } h_N$ and the no. of $\text{NON-EMPTY NODES } n_N$ for RANDOM data of size $N$?
Asymptotic Results for RmST: for large data size $N$

1) Height $H_N$:
   - $\langle H_N \rangle \approx a_m \log(N) + b_m \log(\log(N))$ (??) + …
   - $\text{Var}(H_N) \approx O(1)$

2) Balanced Height $h_N$: Depth upto which the tree is balanced.
   - $\langle h_N \rangle \approx c_m \log(N) + d_m \log(\log(N))$ (??) + …
   - $\text{Var}(h_N) \approx O(1)$

Binary Tree ($m = 2$): $a_2 = 4.31107\ldots$ and $c_2 = 0.3733\ldots$ (Devroye, 87). The correction terms $\to$ conjectured by Hattori and Ochiai (simulations, 2001). Other results by Robson (2001), Reed (2001), Drmota (2001-2003).
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: $\nu_N = \langle (n_N - \langle n_N \rangle)^2 \rangle$.

$$
\nu_N \sim N \quad \text{for } m \leq 26
$$

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

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** $\rightarrow$ **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 $a_m$, $b_m$, $c_m$, $d_m$ **EXACTLY** for all $m$ as roots of transcendental equations. **Scaling Relation** between $a_m$ and $b_m$:

$$\rho_2 = -3a_2/[2(a_2 - 1)].$$

We show that $m_c = 26.0461...$: Find $\lambda(m)$ from $m(m - 1)B(\lambda + 1, m - 1) = 1$. 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. Dean and S.M., 2002).

Various other generalizations: **Vector Data**
The Mapping to a Fragmentation Process

Construction of the Tree $\rightarrow$ Dynamical Fragmentation Process: Split an interval into $(m - 1)$ pieces with the break points chosen randomly. An interval can split iff it contains at least one point.

Ex: Consider the data: $\{6, 4, 5, 8, 9, 1, 2, 10, 3, 7\}$

NOTE:

No. of NONEMPTY nodes $n=7=\text{No. of SPLITTING EVENTS}$
**Fragmentation Process:**

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 $> N_0 = 1$, 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:

Height $H_N$:
- $\text{Prob}[H_N < n] = \text{Prob}[l_1 < 1, l_2 < 1, \ldots \text{ after } n \text{ steps}]$

Balanced Height $h_N$:
- $\text{Prob}[h_N > n] = \text{Prob}[l_1 > 1, l_2 > 1, \ldots \text{ after } n \text{ steps}]$

Number of Nonempty Nodes $n_N (m > 2)$:
- $\text{Prob}[n_N = n] = \text{Prob}[\text{there are } n \text{ SPILLITING EVENTS till the end of the Fragmentation process}].$
Analysis of HEIGHT $H_N$

$$P(n,N) = \text{Prob}[H_N < n] = \text{Prob}[l_1 < 1, l_2 < 1, \ldots \text{ after } n \text{ steps}]$$

```
N

rN

(1-r)N
```

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

---

INCREASING $\log(N)$

TRAVELLING FRONT

---
Travelling Front in Fisher Equation

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

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

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

$$x_f(t) \sim v t + \ldots$$

Q: How to determine the Front Velocity $$v$$?
Kolmogorov’s Velocity Selection Principle:

\[ \phi(x, t) \sim \exp[-\lambda(x - vt)] \]

DISPERSION RELATION: \[ v(\lambda) = \lambda + \frac{1}{\lambda} \]

\[ \rightarrow \text{minimum at } \lambda^* = 1. \text{ For sharp initial condition, } v = v(\lambda^*) = 2. \]

More generally,

\[ x_f(t) \approx v(\lambda^*)t - \frac{3}{2\lambda^*} \log t + \ldots \] (Bramson, Brunet & Derrida, van Saarloos, ...)
Travelling Front Solution to Search Tree Height:

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

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

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

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

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

\[ a_2 = v(\lambda^*) = 4.31107 \ldots \text{ and } b_2 = -\frac{3}{2\lambda^*} = -1.95303 \ldots \]

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:

\[ n(N) \equiv n(r_1 N) + n(r_2 N) + n(r_3 N) + \cdots + n(r_m N) + 1; \quad \sum_{i}^{n} r_i = 1 \]

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:

The marginal distribution of any fragment: \( \eta(r) = (m - 1)(1 - r)^{m-2} \)
Integral Equations for Average and Variance:

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

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

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 $S = \sum_{i=1}^{n} \mu(r_iN)$.

These integral equations can be solved analytically: for large $N$,

$$\nu_N \sim N \quad \text{for } m \leq m_c$$

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

where $m_c$ is determined as:

Find $\lambda(m)$ from $m(m-1)B(\lambda+1, m-1) = 1$. The critical value $m_c$ is obtained by setting, $\text{Re}[\lambda(m) = 1/2]$. For $m > m_c = 26.0461...$, $\theta(m) = \lambda(m)$. (D. Dean and S.M., 2002).
Generalization to Vector Data:

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:

\[ \begin{array}{|c|c|c|}
\hline
N_2 & 7 & 6 \\
\hline
6 & 5 & 4 \\
\hline
5 & 4 & (6, 4) \\
\hline
4 & (3, 3) \\
\hline
3 & 2 & 1 \\
\hline
\end{array} \]

- $\rightarrow$ splitting due to $(6, 4)$
- $\rightarrow$ splitting due to $(4, 3)$
- $\rightarrow$ QUAD-TREE

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

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 $\nu_N$ has a Phase Transition
- $\nu_N \sim V$ for $D \leq D_c$
- $\sim V^{2\theta(D)}$ for $D > D_c$

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

$$\theta(D) = 2 \cos\left(\frac{2\pi}{D}\right) - 1 \rightarrow \text{increases continuously with } D$$

for $D > D_c$
Probability Distribution of the no. of Non-Empty Nodes $n_V$:

$P[n_V] \rightarrow \text{GAUSSIAN \ for \ } D < D_c = 8.69362 \ldots$

$P[n_V] \rightarrow \text{NON-GAUSSIAN \ for \ } D > D_c = 8.69362 \ldots$
Summary and Conclusion:

- Analysis of \( m \)-ary search trees via techniques of statistical physics $\rightarrow$ \textbf{Exact asymptotic results}.

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

- Application of the \textbf{Travelling Front} technique in computer science problem.

- A simple mechanism for the peculiar \textbf{Phase Transition} in the fluctuation of the number of non-empty nodes

$\rightarrow$ A rather \textbf{Generic phase transition} $\rightarrow$ \textbf{New Exact Results for Vector Data}.

The same mechanism is also responsible for the phase transition in a \textbf{Growing Tree Model} of Aldous & Shields (1988)...Explicit Results (S.M. and D.S. Dean, 2004).

\textbf{Perspectives:} Lots of beautiful open problems in \textit{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)