Some Applications of Statistical Mechanics for Searching and Sorting in Computer Science

David DEAN and Satya MAJUMDAR

Laboratoire de Physique Théorique, IRSAMC
Université Paul Sabatier
Toulouse, FRANCE

Plan of talk

- Sorting and Searching for Data
- Height of a Random Binary Search Tree - Directed Polymers
- Number of Nodes of Random m-ary Search Tree - Fragmentation (Dean and Majumdar J. Phys. A. 35, L501, (2002))
- Conclusion

Physicists and computer scientists often look at the same sort of problems.

They don't always ask the same questions !

Sorting and Searching

Sorting and searching with randomized algorithms: how to store data inorder to recover a given data element quickly.

Put an order on the data: data $\rightarrow$ integers

- Days of week $\rightarrow$ 1-7.
- Months of the year $\rightarrow$ 1-12.
- Letters of alphabet $\rightarrow$ 1-26.
- MI5 agents $\rightarrow$ 001 - 0010.

N data elements arriving randomly:$\{6,\ 4,\ 5,\ 8,\ 9,\ 1,\ 2,\ 10,\ 3,\ 7\}$ $N = 10$

Linear sorting and searching

Sorting (storing): in a linear table, put n-th arriving element in box n.

Searching: look in box 1, if required element found stop, if not look in box 2 etc. until you find required element. Characteristic search time $O(N)$

Search trees - exploit the ordering

Binary search tree (BST)

Sorting

- Put element 1, n_1 in the first node of the tree.
- New incoming data moves down the tree, at each occupied node it goes to the bottom right/left if it is bigger/less than the element in the occupied node.
- Data stored when it finds an empty node.

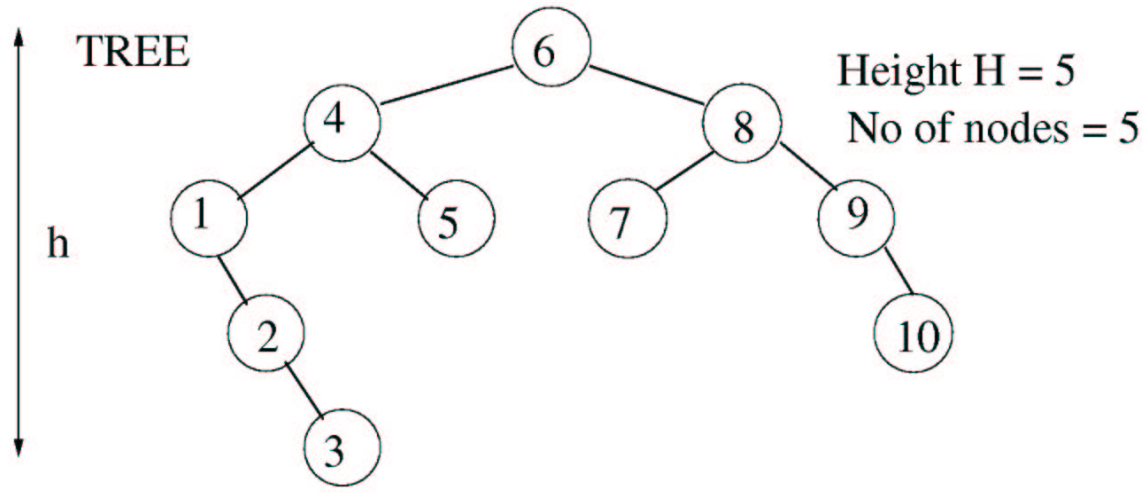
Each data item n stored at a height h_n = search time of element n. Characteristic search time h_c: $2^{h_c} \sim N$. Hence $h_c \sim O(\ln(N))$.

Longest search time $H = \max\{h_n\ 1 \leq n \leq N\}$ = Height of search tree.

Random BST: all data sequences are equiprobable within the $N!$ possibilities. Devroye 1986, Majumdar and Krapivsky 2002 average worst case search time for Random BST: $\langle H \rangle \sim \alpha \ln(N)$ with $\alpha = 4.31107$. Worst case scenario possible $H = N$ (but you'd have to be very unlucky).

Example of a BST

DATA ={ 6,4,5,8,9,1,2,10,3,7}



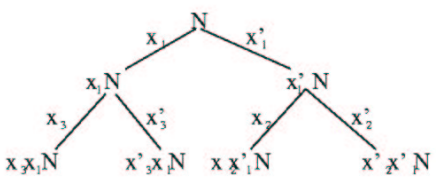
Height of a Random Binary Search Tree

Devroye, Majumdar and Krapivsky: the construction of the binary search tree can be seen as a successive fragmentation of the data set.

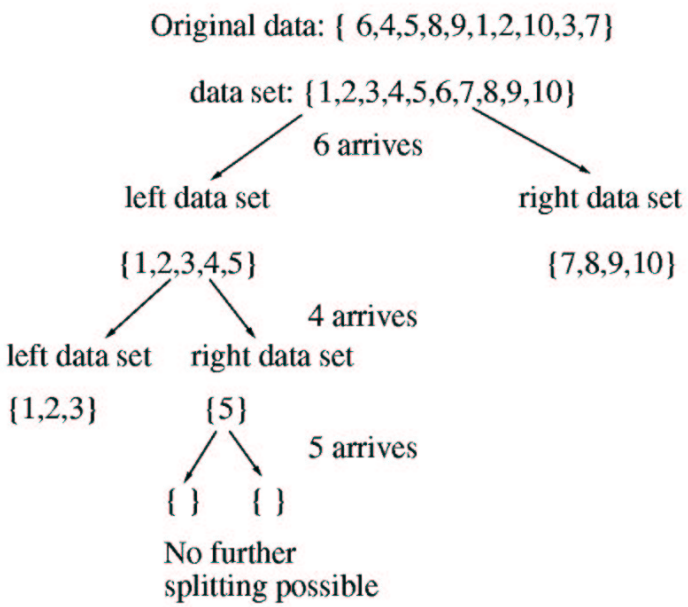
If the first element n_1 is the integer k, then the data set is spit up into a left set $\{1,\ 2, \cdots k-1\}$ of size $k-1$ and a right set $\{k+1,\ k+2 \cdots N\}$ of size $N-k$. These sets are stored to the left and right of the first node and sub-trees they will form will be independent.

Each splitting leads to the creation of a left/right node for each non-empty left/right data set created from the original split data set.

For large N in the Random BST a splitting creates a left data set of size xN and right data set of size $x'N$ with $x' = 1 - x$ and x uniformly distributed on $[0, 1]$



Example of data fragmentation



Mapping onto a Directed Polymer

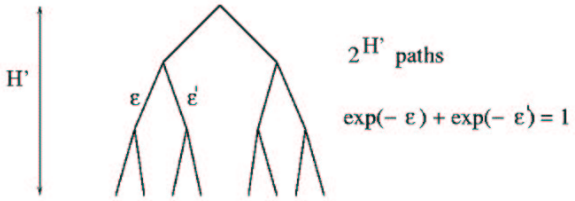
We consider a complete binary tree of height H'. Associate with branch a the splitting fraction x_a.

If two branches a and b have the same parent node then $x_a + x_b = 1$, otherwise no correlations.

There are $2^{H'-1}$ directed paths down the tree terminating at the bottom of the tree with data set size $s_{path}(N, H') = N \times \prod_{path} x$. If all $s_{path}(N, H') < 1$ then the binary tree can not grow any further and in terms of the search tree $H_N < H'$.

$$P(H_N < H') = P(\max\{s_{path}(N, H')\} < 1)$$

Write $x = \exp(-\epsilon)$ for each bond and identify $E_{path} = -\ln(s_{path}(N, H')) + \ln(N)$ as the energy of a directed polymer on a binary tree. Apart from the correlation between ϵ and ϵ' induced, this problem has been studied (Derrida and Spohn).



Hence in the directed polymer setting

$$P(H_N < H') = P(\min\{E_{path}(H')\} < \ln(N))$$

Ground state or zero temperature free energy of directed polymer $E_{GS} = \min\{E_{path}(H')\}$. From thermodynamics we expect that E_{GS} is self averaging and extensive: $E_{GS} \approx H'\epsilon^*$ for large H'.

Hence to leading order

$$\langle H \rangle \approx \ln(N)/\epsilon^*$$

REM-like Replica Solution

Partition Function

$$Z = \sum_{paths} \exp(-\beta \sum_{a \in path} \epsilon_a)$$

In REM-like model the annealed free energy is exact down to T_g where the system freezes. Below T_g: entropy $S = 0$ and energy is constant.

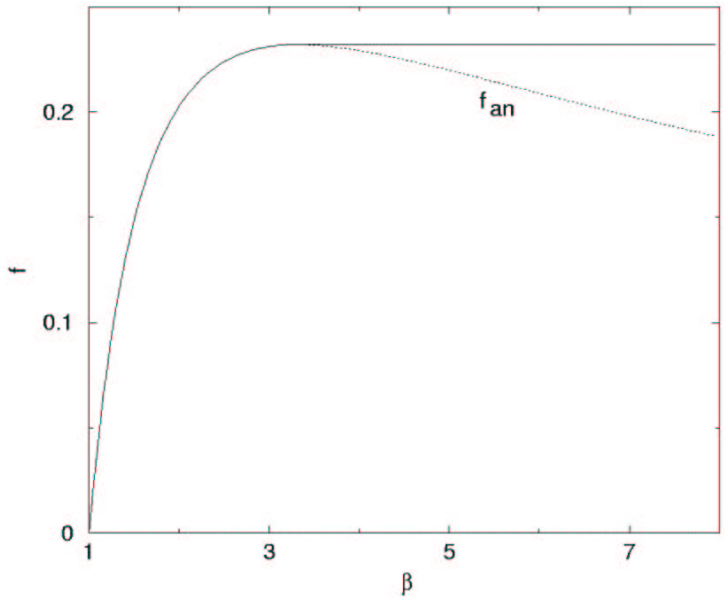
Annealed free energy per height

$$f_{an}(\beta) = -\frac{1}{\beta} \ln \left(2 \int \rho(\epsilon) \exp(-\beta\epsilon) d\epsilon \right)$$

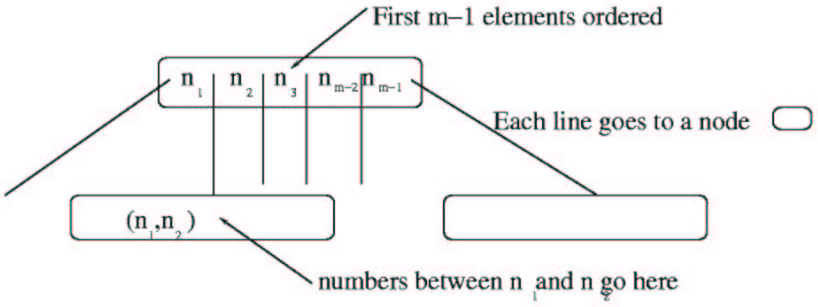
Entropy vanishes at β_g given by $\partial f/\partial\beta = 0$ and $\epsilon^* = f(\beta_g)$. Have $\rho(\epsilon) = \exp(-\epsilon)$ to obtain $\epsilon^* = 1/4.31107$.

More rigorously the problem can be studied by traveling wave techniques which also allow computation of finite size corrections (Majumdar and Krapivsky).

One Step Directed Polymer Free Energy



m-ary Search Trees



Random m-ary search tree: M the number of nodes is random.

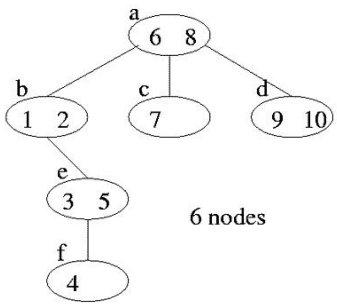
Chern and Hwang: $\langle M \rangle \approx a(m)N \quad var(M) \sim N^{2\theta(m)}$

$\theta(m) = 1/2$ for $m \leq 26$, $\theta(m) > 1/2$ and increasing with m for $m > 26$

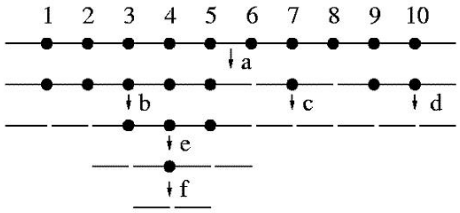
$$a(m) = 1/2\,[\textstyle\sum_{k=2}^{m} 1/k]$$

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

3–ary Search Tree



Interval Splitting (6 splittings)



- For large N the data set is split into data sets of size $Nr_1,\ Nr_2, \cdots Nr_m$.
- Joint density $\eta_m(r_1, r_2, \cdots, r_m) = (m-1)!\delta(\sum_{i=1}^m r_1 - 1)$
- Only data sets containing a point • can split, hence intervals of size < 2 cannot split.
- Each splitting corresponds to creating a node: M = number of splittings.

Generalized fragmentation process

- Start with 'length' x
- Split with distribution η_m.
- Stop when all pieces are smaller than x_0 the atomic size.
- What is $n(x)$ the number of splittings starting from segment of size x ?

$$n(x) = 1 + \sum_{i=1}^m n(r_i x) \quad x \geq x_0; \qquad n(x) = 0 \quad x < x_0$$

Average Number of Splittings

Average $\overline{n}(x) = \mu(x)$ satisfies

$$\mu(x) = 1 + m \int_{x_0/x}^{1} dr\ \eta_1(r)\mu(rx)$$

One point density same for all r_i - isotropic splitting

$$\eta_1(r) = \int\ \eta_m(r, r_2, \cdots r_m) \prod_{i=2}^{m} dr_i.$$

Write $x = e^{\alpha}$ and $\mu(e^{\alpha}) = F(\alpha)$

$$\tilde{F}(s) = \frac{1}{s\,[1 - m w(s)]}, \quad w(s) = \int_0^1 dr\ \eta_1(r)\ r^s$$

Assume $\tilde{F}(s)$ has simple poles at $s = \lambda_k$

$\mu(x) = a_0 + \sum_k a_k x^{\lambda_k}$ with $a_0 = 1/(1-m)$

$a_0 = 1/(1-m)$ from pole at 0 and $a_k = -1/[m\lambda_k w'(\lambda_k)]$.

Length conservation $\sum_i r_i = 1$ implies $\int_0^1 dr\ \eta_1(r)\ r = 1/m$ hence $s = 1$ is always a pole and it is the pole of larges real part.

Let λ and λ^* denote the pair of complex conjugate poles with the next largest real part. Then for large x

$$\mu(x) \approx a_1 x + a_2 x^{\lambda} + a_2^* x^{\lambda^*}$$

$$a_1 = -1/[m \int_0^1 dr\ \eta_1(r) r \ln(r)]$$

The variance

The variance $\nu(x) = \overline{(n(x) - \mu(x))^2}$ obeys

$$\nu(x) = f(x) + m \int_{1/x}^{1} dr\ \eta_1(r)\nu(rx)$$

Same type of equation as before but with different source term $f(x) = \langle (\sum_{i=1}^{m} [\mu(r_i x) - \langle \mu(rx) \rangle])^2 \rangle$.

$$\tilde{\nu}(s) = \frac{\tilde{f}(s)}{[1 - mw(s)]}$$

Large x behavior of $f(x)$ is

$$f(x) \approx b_1 x^{2\lambda} + b_2 x^{2\lambda^*} + b_3 x^{(\lambda+\lambda^*)}$$

2θ = pole of $\tilde{\nu}(s)$ with largest real part: $\theta = \max(1/2, \lambda)$

For large x $\nu(x) \sim x^{2\theta}$: Phase transition when $\mathrm{Re}(\lambda(m)) = 1/2$

Back to the m-ary search tree

- In units of the cut-off $x_0 = 2$ the original size is $x = N/2$.
- $\eta_1(r) = (m-1)(1-r)^{m-2}$

This gives

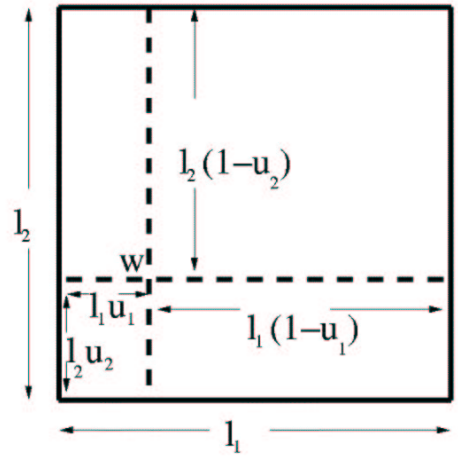
$$p(s) = 1 - mw(s) = 1 - m(m-1)B(s+1, m-1)$$

$p(s)$ has $\mathrm{Re}(\lambda) < 1/2$ for $m < m_c \approx 26.0461$. (no string theory !)

$$a(m) = a_1 \frac{N}{2} = 1/2 \left[\sum_{k=2}^{m} 1/k \right]$$

Phase transition on random m-ary search tree special case of transition in fragmentation problems.

Cuboid splitting



In D dimension: size $x = \prod_{i=1}^{D} l_i$, no of fragments $m = 2^D$.

Computer science link - quadtrees for storing sequences of vectors

- Cut-off atomic size $x_0 = 1$, original size x.
- On splitting $x' = xr(\sigma)$ where
  $r(\sigma) = \prod_{k=1}^{D} (u_k(1+\sigma_k)/2 + (1-u_k)(1-\sigma_k)/2)$, $(\sigma_k = \pm 1)$.
- $\eta_1(r) = \frac{[-\ln(r)]^{D-1}}{(D-1)!}, \quad 0 \le r \le 1.$

$$\tilde{F}(s) = \frac{1}{s\left(1 - \frac{2^D}{(s+1)^D}\right)}.$$

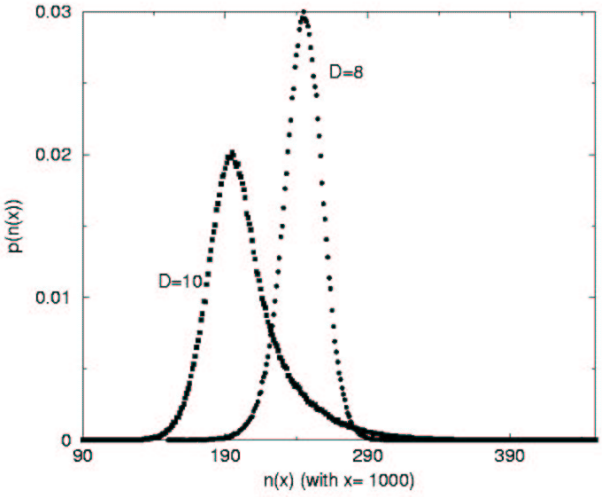
$\mu(x) \approx 2x/D$ for large x.

Poles closest to the left of $s = 1$: $\lambda = -1 + 2e^{2\pi i/D}$ and $\lambda^* = -1 + 2e^{-2\pi i/D}$.

$$\mathrm{Re}(\lambda) = -1 + 2\cos(2\pi/D); \quad D = D_c = \pi/\sin^{-1}(1/2\sqrt{2}) = 8.69\ldots$$

Numerical Cuboid Splitting

Distribution $p(n(x))$ of the number of splittings of a cuboid of original volume $x = 1000$ for $D = 8$ (filled circles) and for $D = 10$ (filled squares).



(Histogram formed by numerically splitting 5×10^5 samples in each case)

Conclusions and outlook

Aspects of search trees can be viewed as a fragmentation process.

Fragmentation well studied in physics

- Energy cascades in turbulence (Greiner et al)
- Rupture process in earth quakes (Newman and Gabrielov)
- Stock market crashes (Sornette and Johansen)
- DNA segmentation algorithms (Galvan et al)
- Jamming in metastability (Desmedt et al)

Found a novel phase transition because a computer science problem lead us to ask a new question about fragmentation.

Outlook: Physical system exhibiting the phase transition, distributions, more complicated algorithms