Based on joint work with Cosmin Bonchiş (UVT& IEAT). Supported by IDEI Grant PN-II-ID-PCE-2011-3-0981 ”Structure and computational difficulty in combinatorial optimization: an interdisciplinary approach”.

Partition into heapable subsequences, heap tableaux and a multivariate extension of Hammersley's process

Gabriel Istrate

West University of Timişoara
and the e-Austria Research Institute

Caution

- This talk: tutorial-style presentation of evolving work, unrefereed/not published results.
- ... tech. details omitted in favor of intuition ...

Nothing presented here (that we did) existed on November 1st 2014.

I have worked to make the results look simpler than they are.
Proofs aren't that simple

Longest Increasing Sequence problem

3 2 5 7 1 6 9

- Given n (integer) numbers $a_1, a_2, \ldots, a_n$ find the longest subsequence (not necessarily contiguous) that is increasing

Longest Increasing Sequence problem

3 2 5 7 1 6 9

- Given n (integer) numbers $a_1, a_2, \ldots, a_n$ find the longest subsequence (not necessarily contiguous) that is increasing
- First year algorithms: Dynamic programming.

Longest Increasing Sequence problem

3 2 5 7 1 6 9

- Given n (integer) numbers $a_1, a_2, \ldots, a_n$ find the longest subsequence (not necessarily contiguous) that is increasing
- First year algorithms: Dynamic programming.
- ... but there exists another (greedy, also first year) algorithm: Patience sorting.

Start (greedily) building decreasing piles. When not possible, start new pile.

Patience Sorting

3 2 5 7 1 6 9

Patience Sorting

3 2 5 7 1 6 9

- 3

Patience Sorting

3 2 5 7 1 6 9

- 3 2

Patience Sorting

3 2 5 7 1 6 9

- 3 2
- 5

Patience Sorting

3 2 5 7 1 6 9

- 3 2
- 5
- 7

Patience Sorting

3 2 5 7 1 6 9

- 3 2 1
- 5
- 7

Patience Sorting

3 2 5 7 1 6 9

- 3 2 1
- 5
- 7 6

Patience Sorting

3 2 5 7 1 6 9

- 3 2 1
- 5
- 7 6
- 9

Longest increasing sequence of a random permutation

QUESTION:

If π is a random permutation in S_n, what is the expected value of $LIS(\pi)$?

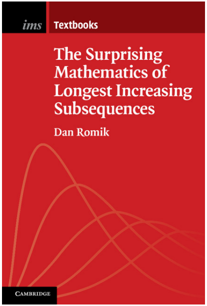
ANSWER:

$$E_{\pi \in S_n}[LIS(\pi)] = 2\sqrt{n} \cdot (1 + o(1)).$$

Logan & Shepp (1977), Veršik-Kerov (1977), Aldous-Diaconis (1995)

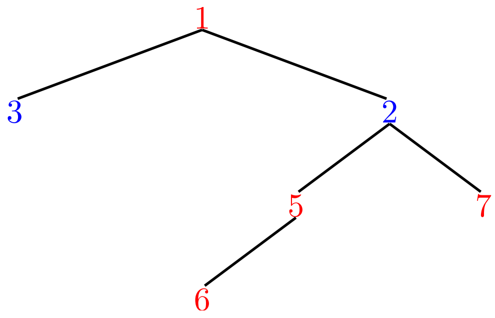
MORE IMPORTANTLY ...

- LIS: very rich problem.
- Connections with the theory of interacting particle systems (nonequilibrium statistical physics)
- Also with the theory of Young tableaux
- What ? ... more explanations later.



Some more first-year algorithmics

- Min-Heap: binary tree (usually complete), holds numbers.
- $A[parent(x)] \leq A[x]$.
- Usually have to rearrange numbers into heap: HEAPIFY

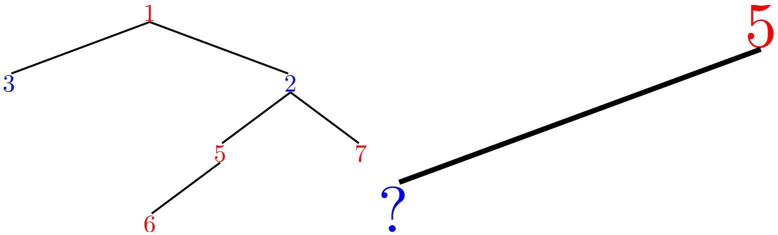


From data structures to sequences

Sequence of integers A is heapable if it can be inserted into binary heap-ordered tree (not necessarily complete) without having to call HEAPIFY.

Byers, Heeringa, Mitzenmacher, Zervas ANALCO'2011

Example: 1 3 2 7 5 6   Counterexample: 5 1 . . .



Earlier results on heapability

- Polynomial time greedy algorithm to decide heapability
- but complete heapability NP-complete
- Longest Heapable Subsequence (LHS): don't know its complexity ...
- But with high probability $LHS(\pi) = n - o(n)$, where $\pi \in S_n$ is a random permutation.

$\boxed{\text{Recall that } E[LIS(\pi)] \sim 2\sqrt{n}!}$

Byers, Heeringa, Mitzenmacher, Zervas ANALCO'2011

Intuition

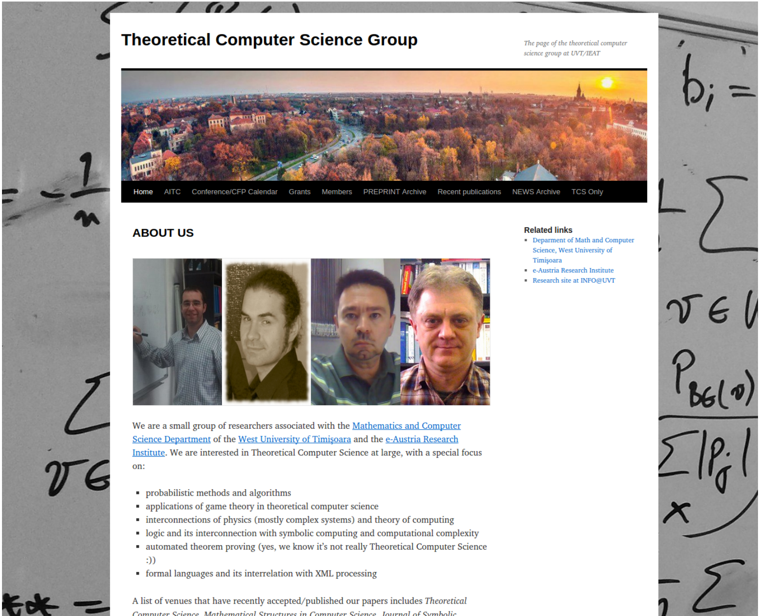
heapable sequences: ”weak” versions of increasing sequences.
Recall 1 3 2 7 5 6

- Question (BYERS et al.): LIS relates to lots of wonderful things
- . . . does LHS relate as well ?

Our answer: to many (at least)

And now, a word from our sponsors ...

Webpage Screenshot



Theoretical Computer Science Group

The page of the theoretical computer science group at UVT/IEAT

Home AITC Conference/CFP Calendar Grants Members PREPRINT Archive Recent publications NEWS Archive TCS Only

ABOUT US

We are a small group of researchers associated with the Mathematics and Computer Science Department of the West University of Timişoara and the e-Austria Research Institute. We are interested in Theoretical Computer Science at large, with a special focus on:

- probabilistic methods and algorithms
- applications of game theory in theoretical computer science
- interconnections of physics (mostly complex systems) and theory of computing
- logic and its interconnection with symbolic computing and computational complexity
- automated theorem proving (yes, we know it's not really Theoretical Computer Science :))
- formal languages and its interrelation with XML processing

A list of venues that have recently accepted/published our papers includes *Theoretical Computer Science, Mathematical Structures in Computer Science, Journal of Symbolic*

Related links

- Deparment of Math and Computer Science, West University of Timişoara
- e-Austria Research Institute
- Research site at INFO@UVT

Back to patience sorting

3 2 5 7 1 6 9

- 3 2 1
- 5
- 7 6
- 9

- Partitions the array into decreasing (down-)sequences
- SDS(A): the minimal number of such subsequences.
- *... (digression): measure of "sortedness of sequence A."*

THEOREM(classical): $SDS(A) = LIS(A)$, and patience sorting computes $SDS(A)$.

''Patience heaping''

k-heapable: heapable into a k-ary min-heap
$MHS_k(A)$**:** the smallest number of k-heapable subsequences in a decomposition of A.
Slot of a node: the k (free) positions where children may grow.

Algorithm 1.1: PATIENCE-HEAPING(W)

INPUT $W = (w_1, w_2, \ldots, w_n)$ a list of integers.
Start with empty heap forest $T = \emptyset$.
for i in range(n):
 if (there exists a slot where X_i can be inserted):
  insert X_i in the slot with the lowest value.
 else :
  start a new heap consisting of X_i only.

”First (easy) result”

THEOREM 1: ”Patience heaping” computes $MHS(A)$.

Proof Idea:

- Define domination relation between multisets of slots.
- A dominates B: A ”grows slower than B”.
- ”Greedy insertion steps dominate any other insertion steps”
- induction ...

If GREEDY creates new heap then any other algorithm does.

”Second (easy) result”

THEOREM 2: The following statements are true for every $k \geq 2$:

(a). there exists a sequence X such that

$$MHS_k(X) < MHS_{k-1}(X) < \ldots < MHS_1(X).$$

(b). $\sup_X \left[MHS_{k-1}(X) - MHS_k(X)\right] = \infty.$

Proof Idea:

Examples . . .

A beautiful conjecture (I)

How does $E[MHS_k(\pi)]$, where $\pi \in S_n$ is a random permutation, grow ?

- Increasing $\equiv$ "1-heapable". $E[LIS(A)] \sim 2\sqrt{n}$.
- For any k growth at least logarithmic.

THEOREM 3: For every fixed $k, n \geq 1$

$$E_{\pi \in S_n}[MHS_k(\pi)] \geq H_n = 1 + \frac{1}{2} + \ldots + \frac{1}{n}, \tag{1}$$

the n'th harmonic number, $\sim \ln(n)$.

A beautiful conjecture (II)

PROOF IDEA:

- Every left-to-right minimum in the permutation starts a new heap.
- First element: minimum with probability 1.
- Second: minimum with probability $\frac{1}{2}$.
- ...
- n'th (last): minimum with probability $\frac{1}{n}$.
- linearity of expectation ....

Conjecture: for $k \geq 2$ the expected growth is logarithmic. More precisely ...

A beautiful conjecture

CONJECTURE: We have

$$\lim_{n\to\infty} \frac{E[MHS_2[\pi]]}{\ln(n)} = \phi,$$

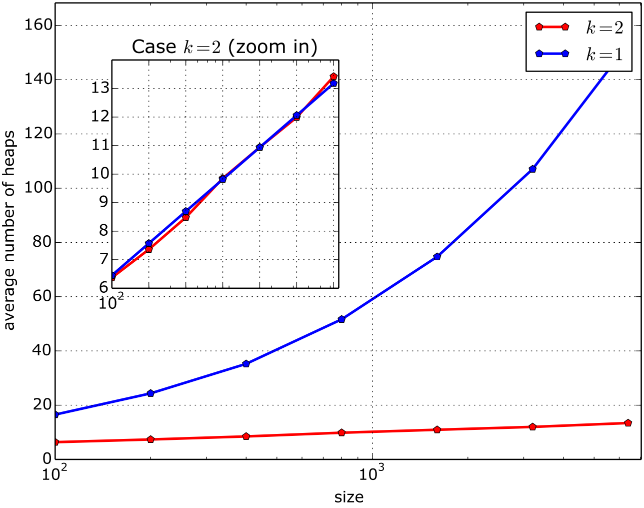
with $\phi = \frac{1+\sqrt{5}}{2}$ the golden ratio.

More generally, for an arbitrary $k \geq 2$ the relevant scaling is

$$\lim_{n\to\infty} \frac{E[MHS_2[\pi]]}{\ln(n)} = \frac{1}{\phi_k}, \tag{2}$$

where ϕ_k is the unique root in $(0, 1)$ of equation
$X^k + X^{k-1} + \ldots + X = 1.$

Evidence we have: I. Simulations.



Evidence we have

- We kind of know what's going on.
- Can make nonrigorous computations that match experimental predictions
- . . . like physicists do !

We just can't make all steps of the argument rigorous !

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3

0.3

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2

0.2 ~~0.3~~

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2
- 0.5

0.2 0.5

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2
- 0.5
- 0.7

0.2 0.5 0.7

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2 0.1
- 0.5
- 0.7

- 0.1 ~~0.2~~ 0.5 0.7

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2 0.1
- 0.5
- 0.7 0.6

- 0.1 0.5 0.6 $\cancel{0.7}$

Physics of patience sorting

0.3 0.2 0.5 0.7 0.1 0.6 0.9

- 0.3 0.2 0.1
- 0.5
- 0.7 0.6
- 0.9

- 0.1 0.5 0.6 0.9

Physics of patience sorting (II)

Hammersley's process:

- Particles arive at integer time as random real numbers $X_i \in [0, 1]$.
- Particle X_i kills closest live particle X_j, $X_i < X_j$ (if any)

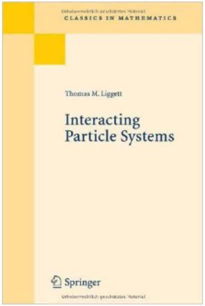
At time n the number of piles in patience sorting is equal to the number of live particles in Hammersley's process.

- studied in the area of interacting particle systems, a field between probability theory and (Nonequilibrium) Statistical Physics.
- relative of a more famous process, the so-called Totally Asymmetric Exclusion Process (TASEP)

Hammersley's process: an interacting particle system.

- Most illuminating proof of $E[LIS(\pi)] \sim 2\sqrt{n}$ via the analysis of the so-called hydrodynamic limit of Hammersley's process.
- Can be embedded into an (asymptotic) Poisson process in $\mathbf{R}^2$.
- Physicists can analyze via PDE without worrying about rigor.

$$\frac{\partial F}{\partial t} = 1/\frac{\partial F}{\partial x}, F(t,0) = F(0,x) = 0.$$



Physics of patience heaping ?

Hammersley's process with k lifelines (HAM_k):

- "Particles" arrive at integer times as random numbers $X_i \in [0, 1]$.
- each "particle" is initially endowed with k lifelines.

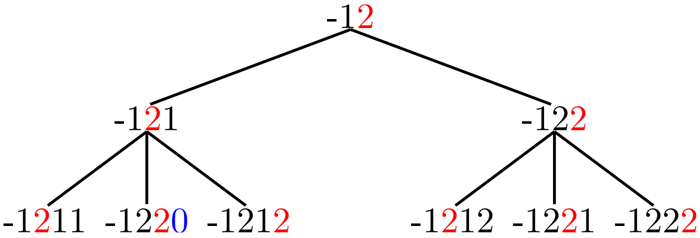
- X_i ~~kills~~ takes one lifeline from closest live X_j, $X_i < X_j$ (if any)

THEOREM 4: At time n the multiset of free slots in patience heaping corresponds (with multiplicities) to live particles in Hammersley's process with k lifelines.

number of heaps: increases by one when the new particle doesn't take any lifeline (maximum).

How to simulate process HAM$_k$

- Words over alphabet $0, 1, 2$ and a (conventional leading) -1.
- Start with $W_0 = -1$ (leftmost marker)
- Choose a random position to the right of -1. Put there a 2. Remove 1 from the closest nonzero digit to the right (if any).



Towards analyzing process HAM$_k$

$d_0(n), d_1(n), d_2(n)$: average densities of digits 0,1,2 in word W_n (discarding -1).

THEOREM 5: There exist constants $d_0, d_1, d_2 \in [0, 1]$ such that

$$\lim_{n \to \infty} d_i(n) = d_i, i = 0, 1, 2.$$

Proof Idea:

Fekete's Lemma (subaditivity): "If a_n is a sequence with $a_{m+n} \le a_m + a_n$ then $\lim_{n \to \infty} a_n n$ exists."

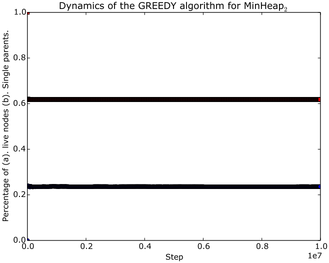
Apply Fekete's lemma to some suitable independent linear combinations of $d_i(n)$.

''A physicist like's explanation for the dynamics of HAM$_k$''

- Empirically $d_1 + d_2 = \frac{\sqrt{5}-1}{2} \sim 0.618\ldots$, $d_2 = \sqrt{5} - 2 \sim 0.236$ and of course $d_0 + d_1 + d_2 = 1$.

Five independent runs, each with 10.000.000 simulation steps. The final values are listed in Figure 1. The first trajectory itself, is plotted in Figure 2, albeit only once every 10.000 steps (remember, we make 10.000.000 moves !). Basically we've converged very fast.

```
>>> ============== RESTART ========
>>>
final[ 1 ]: 0.6179875 0.2359724
final[ 2 ]: 0.6180139 0.2360253
final[ 3 ]: 0.6180207 0.236038
final[ 4 ]: 0.61804 0.2360777
final[ 5 ]: 0.6180516 0.2361006
```



''A physicist like's explanation for the dynamics of HAM$_k$ (II)''

- In the limit of $n \to \infty$ string W_n looks like a random string of 0,1,2 (with the given densities)
- ... this is because the limit should be described by a compound Poisson process
- Assuming well mixing of digits one can write evolution equations that predict values of d_0, d_1, d_2.

”A physicist like’s explanation for the dynamics of HAM_k (III)”

- Assuming well mixing, largest nonzero digit (there are $\sim \frac{\sqrt{5}-1}{2}n$ of them) ”has rank $\sim (n - \frac{\sqrt{5}+1}{2})$”
- The probability that the new particle at time n come above this element, therefore increasing the number of heaps, is $\sim \frac{\sqrt{5}+1}{2n}$.
- Thus $E[MHS_2(\pi)]$ is $\sim \frac{\sqrt{5}+1}{2} \cdot H_n \sim \frac{\sqrt{5}+1}{2} \ln(n)$.

Will explain/substantiate all these in a second, ”physics-like” paper.

Digression: A reaction to our work

- Michael Mitzenmacher, professor of Computer Science at Harvard, ”leading expert in hash function applications such as Bloom filters, cuckoo hashing, and locality sensitive hashing.” (Wikipedia)
- One of the authors of the original ”heapable paper”.

I love the new conjecture that the expected minimal number of heapable subsequences a random sequence decomposes into is ((1+sqrt{5})/2) ln n. (It's clearly at least ln n, the expected number of minima in the sequence.)

There are still all sort of open questions, that seem surprisingly difficult; and I certainly can't claim I know of any important practical applications. But Longest Heapable Subsequences just appeal to me as a simple, straightforward mathematical object that I wish I understood more.

For simple-sounding but apparently difficult open questions, as far as I know, the answer to even the basic question of "What is the formula for how many sequences of length n are heapable?" is still not known. Similarly, I think the question of finding an efficient algorithm for determining the Longest Heapable Subsequence (or showing it is hard for some class) is open as well.

POSTED BY MICHAEL MITZENMACHER AT 6:45 PM 2 COMMENTS:
LINKS TO THIS POST

BLOG ARCHIVE

▼ 2015 (1)
▼ February (1)
New Heapable Subsequence Paper
► 2014 (42)
► 2013 (70)
► 2012 (81)
► 2011 (78)

Heapable sequences and Young tableaux

- Patience sorting related to Young tableaux
- Useful in pure math and theoretical physics.
- Robinson-Schensted-(Knuth): algorithm to create Young tableau from permutation. First row: Hammersley's process

Is there some notion of "heap tableau" that is to heapable sequences/HAM$_k$ process what Young tableaux are for longest increasing sequence ?

- Two ways. we did the easier version. Only partial success.
- More work needed

Hook formula for Young tableaux

1	3	5
2	6	
4		

5	3	1
3	1	
1		

Figure : (a). Young tableau (b). hook lengths.

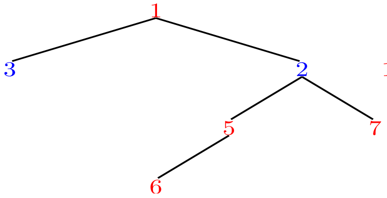
- How many ways to fill up a Young tableau with n cells with numbers from 1 to n ?
- Hook length of cell: number of elements to the right or below it.

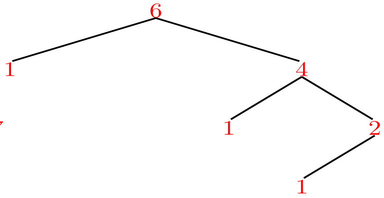
Hook formula for Young tableaux (Frame, Robinson, Thrall'54):

$$\frac{n!}{\prod_i H_i}$$

Proof: nontrivial.

Hook formula for heaps





- Hook length of node: number of nodes <u>at or below</u> it.

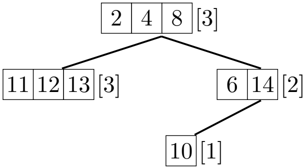
<u>Hook formula for heaps</u> (Knuth, TAOCP exercise in vol. 3):

$$\frac{n!}{\prod_i H_i}$$

<u>Proof:</u> Each node is the smallest in its hook independently with probability $1/H_i$.

Heap tableaux

Is there a common generalization of hook formulas for heaps/Young tableaux ? Is there an object that generalizes both heaps and Young tableaux ?



	λ	0	1	00	01	10
1	2	11	6	$\perp$	$\perp$	10
2	4	12	14			
3	8	13				

	λ	0	1	00	01	10
1	6	3	3	$\perp$	$\perp$	1
2	4	2	1			
3	2	1				

Figure : (a). Heap tableau T_1 and its shape $S(T_1)$ (in brackets) (b). The equivalent Young tableau-like representation of T_1 (c). hook lengths.

Hook inequality for heap tableaux

- Our heap tableaux generalize both heaps and Young tableaux.
- not the version of heap tableaux that corresponds to HAM_k
- Hook formula works for heaps and Young tableaux.
- ... but fails for general heap tableaux.
- perhaps other notion of hook length will do.

THEOREM 6: Given $k \geq 2$ and a k-shape S with n free cells, the number of ways to create a heap tableau T with shape S by filling its cells with numbers $\{1, 2, \ldots, n\}$ is at least

$$\frac{n!}{\prod_{(\alpha,i)\in dom(T)} H_{\alpha,i}}.$$

The bound is tight for Young tableaux, heap-ordered trees, and infinitely many other examples, but is also **not** tight for infinitely many (counter)examples.

Extending the Robinson-Schensted bijection to heap tableaux

I AM OMITTING HERE THE PRESENTATION OF AN ALG. FROM THE PAPER.

- Robinson-Schensted correspondence: bijection between permutations in S_n and pairs of tableaux with n cells of the same shape.
- With an additional twist it continues to hold ...

THEOREM 7: For every $k \geq 2$ there exists a bijection between permutations $\pi \in S_n$ and pairs (P, Q) of k-heap tableaux with n elements and identical shape, where Q is a standard tableau.

"Q is standard" specific to $k \geq 2$: heaps have "too many degrees of freedom" between siblings. An alg. chooses one specific way.

I know, a lot of information for one talk ...

- We've only scratched the surface
- A lot more to do.
- If interested, talk to us.

Thanks !

{gistrate,cbonchis}@info.uvt.ro