6.851	Class 1	Feb. 5, 2014

6.851: Advanced Data Structures

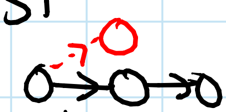
Prof. Erik Demaine
TAs: Tim Kaler & Aaron Sidford
http://courses.csail.mit.edu/6.851/spring14

Technical overview:

Themes:

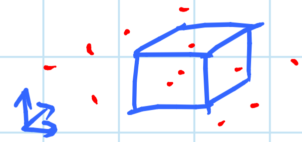
- models of computation: matter!
- fancy data structures: cool!
- tight lower bounds: hard!

Temporal DS: manipulate time (time travel)

- persistence: fixed past
  - motivation: undo, geometry (time=space)
  - partial: linear time, query past
  - full: branching time ⇒ tree 
  - ↳ generally possible with O(1) overhead (partial & full)
  - confluent: can merge timelines ⇒ DAG
  - ↳ lots of results & open problems
    e.g. confluent files & directories solved in 851!
- retroactivity: change the past
  - motivation: mistake correction, geometry
  - hard in general

Geometric DS: points in $d > 1$ dimensions

- motivation: relational databases
- can preprocess n points in 3D to find all points in query box in $O(\lg n)$ time



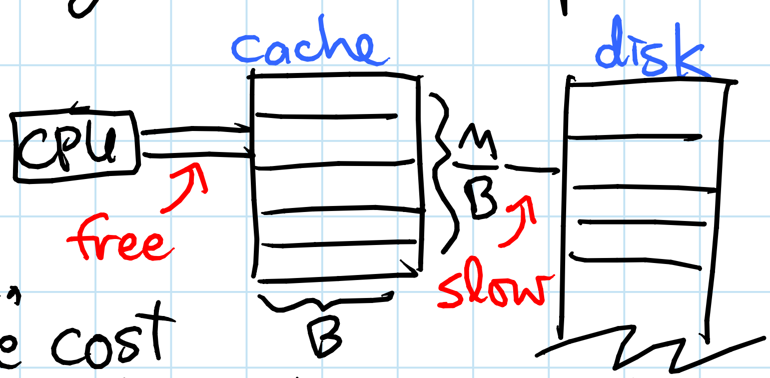
- kinetic DS: moving points

Dynamic optimality: is there one best BST?

O(1)-competitive against any BST?

- any balanced BST is $O(\lg n)$-compet.
- Tango Trees are $O(\lg \lg n)$-competitive
- conjecture: Greedy is $O(1)$-competitive

Memory hierarchy:



- when you load 1 word of data, get B for same cost
- goal: amortize high cost over B items
- scanning N items costs $\Theta(\lceil N/B \rceil)$
- sorting N items costs $\Theta\left(\frac{N}{B} \log_{M/B} \frac{N}{B}\right)$ e.g. $\frac{M}{B}$-way mergesort

& priority queue in $\Theta\left(\frac{1}{B} \log_{M/B} \frac{N}{B}\right)$ $< 1!$

- can do all this without even knowing B & M! "cache oblivious"

$\Rightarrow$ works well on multilevel hierarchy too

Integer DS: words store ints $\in \{0, 1, \ldots, u-1\}$
↳ w bits $\Rightarrow u = 2^w$

- hashing is one example:
  $O(1)$ time w.h.p. insert/delete/search
- insert/delete/predecessor/successor
  (like BSTs): for $O(n \text{ polylg } n)$ space,
  started in 851! → $\Theta\left(\min\left\{\log_w n, \frac{\lg w}{\lg \frac{\lg w}{\lg\lg n}}\right\}\right) \le \begin{cases} O(\sqrt{\lg n}) \\ O(\sqrt{\lg w}) \end{cases}$
- sorting in $O(n)$ time / $O(1)$ priority queue
  for $\underbrace{w = O(\lg n)}_{\text{radix sort}}$ & $w = \Omega(\lg^{2+\varepsilon} n)$

String DS: preprocess text T to search for substring P in $O(|P|)$ ← indep. of T!

- find longest common prefix of 2 (preprocessed) strings in $O(1)$ time

Succinct DS: above in $O(|T|)$ bits, not words

- store n parentheses in $n + o(n)$ bits
  & find matching/parent parens in $O(1)$ time

Dynamic graphs: insert/delete edges
& query: are v & w connected via path?

- $\Theta(\lg n)$ for trees (Ω solved in 851!)
- $O(\lg n \cdot (\lg\lg n)^3)$ for undirected graphs
- we'll see $O(\lg^2 n)$

Class format:

- video lectures from 2012 ← required attendence
  - completion & feedback form
    DUE TUESDAYS AT NOON
- Piazza for raising questions
- class (W2:30-5) for every 2 lectures ← required attendence
  - Q & A (based on form/Piazza feedback)
  - group puzzle solving
    ↳ build collaboration skills, warmup for:
  - attack open problems
    ↳ build research skills, thrill of unknown, fun & challenge of advancing frontiers of research
- weekly psets: 1 page in, 1 page out
  USUALLY DUE MONDAYS AT NOON
- final project: written & presented
  - pose and/or try to solve open problem
    (e.g. from open problem session)
  - implement & experiment with DS
  - survey a few papers (not well-covered)
  - improve Wikipedia

  PROPOSAL DUE APRIL 9, 2014

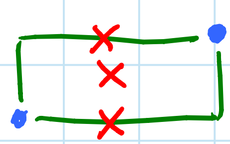
Problems: groups of 5-10 people

- don't worry about solving ~ about journey
- solved problems: write progress in Piazza private note ~ we'll post answers for record
- when done/bored, move on to open problem

Problem 1: insert/delete/successor/pred. in $O(\lg n)$
+ insert-after/before in $O(1)$ amortized
↳ given node x e.g. found by pred./succ., insert $y = x \pm \varepsilon$
[+ delete-here in $O(1)$ amortized
↳ given node x, delete it]
↳ interesting with or without this

Problem 2: insert/delete/pred./successor in $O(\lg n)$
[+ split in $O(\lg n)$]
↳ given DS & key x, split DS into DS of all items $\le x$ & DS of all items $> x$
[+ concatenate in $O(\lg n)$]
↳ given 2 DSs A & B such that $a < b$ for all $a \in A$ & $b \in B$, combine into 1 DS of all items in $A \cup B$
→ either op. is interesting

Open problem: given n points P in 2D, no 2 on common row [or column], find minimum point set $Q \supseteq P$ such that: for any 2 points $\in Q$ not on common row/col., the rectangle they span contains another point $\in Q$



- NP-hard?
- $O(1)$-approximation?