6.851	Class 1	Sept. 6, 2017

6.851: Advanced Data Structures

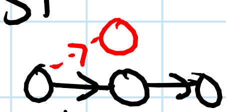
Prof. Erik Demaine
TAs: Adam Hesterberg & Jayson Lynch
http://courses.csail.mit.edu/6.851/fall17/

Technical overview:

Themes:

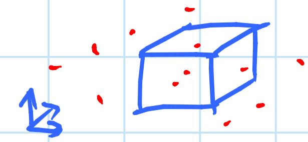
- models of computation: matter!
- fancy data structures: cool!
- tight lower bounds: hard!

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

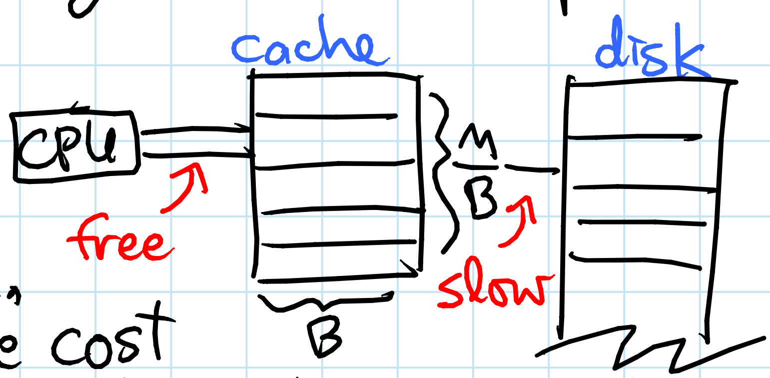


- kinetic DS: moving points

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 $\Theta(\frac{N}{B} \log_{M/B} \frac{N}{B})$ e.g. $\frac{M}{B}$-way mergesort

& priority queue in $\Theta(\frac{1}{B} \log_{M/B} \frac{N}{B})$ < 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 $O(n \operatorname{polylg} n)$ 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 + handwritten & scribe notes (from 2012)
  - DEMO: note sync & jump, playback speed
  - completion & scribe feedback form DEMO
    DUE BY NOON ON TUESDAY
  - questions on Coauthor DEMO
- class (W7–9:30) for every 2 lectures
  (REQUIRED ATTENDANCE)
  - Q&A / additional material (based on feedback)
  - group puzzle solving "SOLVED"
    ↳ learn material, collaboration skills, warmup for:
  - attack open problems "OPEN"
    ↳ build research skills, thrill of unknown, fun & challenge of advancing frontiers of research
  - implement data structures "CODING"
- Coauthor software to coordinate in/outside class
  MUST POST/BE @MENTIONED EACH WEEK DEMO STATS
- weekly psets: 1 page in, 1 page out DEMO PS1
  USUALLY DUE WEDNESDAYS AT NOON
- scribe notes: revise/improve old notes including students' feedback
  (DUE TUESDAY NOON FOLLOWING CLASS)
- project & presentation
  - pose and/or try to solve open problem (e.g. from class)
  - implement & experiment with DS
  - survey a subfield (not well-covered in lectures)
  - Wikipedia (write/improve several articles)