6.851	Class 1	Feb. 18. 2021

6.851: Advanced Data Structures

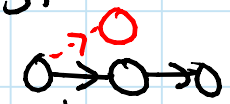
Prof. Erik Demaine
TAs: Josh Brunner, Yevhenii Diomidov, Dylan Hendrickson
http://courses.csail.mit.edu/6.851/spring21/

Technical overview:

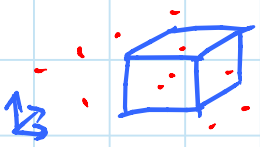
Themes:

- 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
  - 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

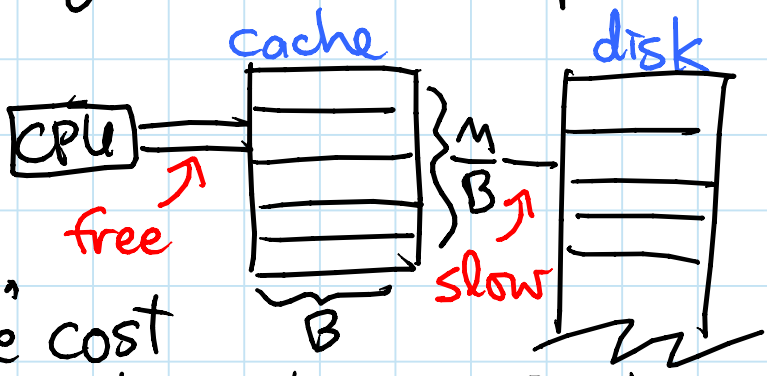
Geometric DS: points in $d > 1$ dimensions
- motivation: relational databases
- can preprocess n points in 3D to find all points in query box in $O(\lg n)$ time



- 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 $O(n \operatorname{polylg} n)$ space,
  started in 851! → $\Theta\left(\min\left\{\log_w n, \frac{\lg w}{\lg \frac{\lg w}{\lg\lg n}}\right\}\right) \leq \begin{cases} O(\sqrt{\lg n}) \\ O(\sqrt{\lg w}) \end{cases}$
- sorting in $O(n)$ time / $O(1)$ priority queue for $w = O(\lg n)$ & $w = \Omega(\lg^{2+\varepsilon} n)$
  radix sort

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 $O(\lg^2 n)$

Requirements: (taking for credit)

Measurement: completion 👍 on Coauthor

- watch all video lectures (2 × 1.5 hr/week / speed) — 20% REQUIRED
  & recorded portions of nonattended synchronous:
- synchronous sessions M4 & R7 (1.5 hr+) — 40% REQUIRED
  - attend ≥1, ideally 2/week (anti-calendar-aligned)
  - participate in problem solving
    - solved problems (pset puzzles)
    - open problems (research!)
    - coding problems (theory → practice)
    - collaborate!

    Coauthor ≥1 post on Coauthor by next Thursday
- problem sets, weekly — 40% OPTIONAL
  - one page in, one page out

  Gradescope
- project — 40% OPTIONAL
  - based on in-class work on open/coding probs.
  - can be collaborative & done at any time

  code, write-up, present

↳ TOTAL: 140%

Grading scheme:

- each component graded on 100-point scale, weighted by % above, and summed
  ⇒ can skip project OR problem sets OR do both to make up for lost points
- this decides your letter grade:
  we may use other schemes for +/-

A ≥90
B ≥80
C ≥70
D ≥60
F <60

SOFTWARE: attempt to simulate in-person room full of tables & whiteboards

DOCS ON GITHUB

Comingle: multiroom meetings + glue together tools
- set your name (we'll be tracking attendence)
- tab switching
- drag tabs to customize layout (for your screen)
- maximize tabset button
- open in separate browser tab (e.g. for tablet)
- reload tab button (e.g. if Jitsi isn't working)
- hand raising (request staff presence)
- star room to indicate interest
- switch/leave room - split room (if too big)
- adding tabs (Wikipedia & some other sites)

Coauthor: master record of notes, ideas, progress
- anything worth saving should end up here.
- can use asynchronously too (between meetings)
- questions thread, including completion 👍
- solved problems:
  - your posts are private (to avoid spoiling)
  - we may publish your answers to class
- add whoever you're working with as coauthors

Cocreate:
- multiple pages for fresh thoughts (& performance)
- download SVG to post good stuff to Coauthor
- embed images from Coauthor or Imgur