6.851	Students II	May 11, 2010

Eric Liu & Tom Morgan:

THEORY/SIMPLIFY

sorting/priority queue equivalence

- Thorup transforms $O(n\,S(n))$ sorting alg. into $O(S(n))$ priority queue
  [Thorup 2007: Lecture 12]
  - group list into blocks of size $\Theta(\lg n)$
  - exponentially space out $\Theta(\lg n)$ geometrically increasing buffers
  - atomic heap over the buffers
  - maintain these properties over updates
- can be lazier: flush buffers only when head shrinks too much
- working toward $O(1)$ amortized Decrease-Key

Jingjing Liu: LCAs in DAGs

SURVEY

- LCA(x,y) = common ancestor z of x & y s.t. no descendant of z is common ancest.
- no longer unique
- one example: common ancestor of max. depth
  length of longest path
- if just want representatives, can compute all-pairs LCAs in $O(n^{2.688} \text{ polylog } n)$ time via approx. shortest paths + search, & in $O(nm)$ time via "list merging", etc.
  #edges
- fully dynamic: update = change edges around vx
- Eckhardt et al. 2007: $O(n^{2.5})$ updates, $O(1)$ query
- Kowaluk et al. 2008: path cover method
- as hard as dynamic transitive closure

- OPEN: bounded in/out degree?
  approximation algorithms?
  shortest-distance LCA?

Kevin Kelley & TB Schardl:

THEORY + EXPERIMENT

models of GPU

- kernel = "inside of loop"
  written in extension of suboset of C
- TPUs, SMs, SPs, half-warps, ...
- 6 types of memories, mostly explicitly man.
- coalescing of structured memory reads
  ⇒ many reads for price of one

- model = fork-join parallelism + external mem.
  + no branching
- analyzed matrix transpose, matrix multiply
- some crazy behavior in experiments
  depending on divisibility of problem size
- still work to do

Nicholas Zehender:

THEORY

faster functional tries

- $O(\min\{d \lg \Delta, \lg n\})$!
- cf. $O(d \lg \Delta)$ & $O(\lg n)$ previous bounds
  [Demaine, Langerman, Price : Lecture 19]
- key idea: modify weights $w \to w'$ in globally biased trees

so $W/w' = O(W/w) \Rightarrow$ still $O(\lg n)$
& $W/w' = O(d)$

Mark Chen & Haitao Mao:

EXPERIMENTS

performance of integer DSs

- van Emde Boas: lots of space overhead, even if you store only nonempty parts...
  (but without hash table optimization)
- x-fast tree: build time $O(nw)$ was slow
- y-fast tree: much better
- fusion trees: not useful on 32-bit words (degenerates to slow BST)
  - estimate $w \approx 1000$ to be good?
- exponential trees: simple
- signature sort: still in progress
  again really needs large words $w = \Omega(\lg^{2+\varepsilon} n)$
- timing data has lots of spikes w.r.t. n
- many DSs run out of memory (on 1GB)