6.851	Lecture 5	Feb. 17, 2010

Kinetic data structures: moving data

- think: tracking physical objects (phones, cars, ...)

[Basch, Guibas, Hershberger – J. Alg. 1999]

Data: value/coordinate = (known) function of time (instead of a single number)

- e.g. affine $\underbrace{a}_{\text{initial}} + \underbrace{b}_{\text{velocity}} t$
- bounded-degree algebraic $a + bt + ct^2 + \cdots$
- pseudo-algebraic: any certificate of interest flips true/false $O(1)$ times

Operations:

- modify(x, f(t)): replace x's function
- idea: motion estimation accurate "for a while"
- advance(t): go forward in virtual time
- other updates/queries usually about present (virtual) time

Approach:

- store data structure accurate now
- augment with certificates: conditions under which DS is accurate, which are true now
- compute failure time for each certificate
- store them in a priority queue
- as certs. invalidate, fix DS & replace certs.

Kinetic predecessor:

- want pred./succ. search in present in $O(\lg n)$
- let's try a BST
- certificates: $\{x_i \le x_{i+1}\}$
  where $x_1, x_2, \ldots, x_n$ is an in-order traversal
- $\text{failure}_i = \inf\{t \ge \text{now} \mid x_i(t) \ge x_{i+1}(t)\}$
  (next time certificate i will fail)
- advance(t):
  - while $t \ge$ Q.min:
    - now = Q.min
    - event(Q.delete-min)
  - now = t
- event ($x_i \le x_{i+1}$): (in fact, $x_i = x_{i+1}$ now)
  - swap x_i & x_{i+1} in BST
  - add certificate $x_i' \le x_{i+1}'$
  - replace certificate $x_{i-1} \le x_i$ with $x_{i-1} \le x_i'$
    & certificate $x_{i+1} \le x_{i+2}$ with $x_{i+1}' \le x_{i+2}$
  - update failure times in priority queue

Metrics:

above:

1. **responsive**: when certificate expires (event), can fix DS quickly $\boxed{O(\lg n)}$
2. **local**: no object participates in many certs. $\Rightarrow$ modify is fast $\boxed{O(1)}$
3. **compact**: # certs. is small $\boxed{O(n)}$ $\Rightarrow$ low space
4. **efficient**:

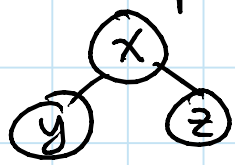
$$\frac{\text{worst-case \# DS events}}{\text{worst-case \# "necessary changes"}} \text{ is small } \boxed{O(1)}$$

Efficiency: (the vaguest part of kinetic DSs)

- if we need to "know" sorted order at all times, need to update for each order change & that's what we do
- if we need to support fast pred./succ. at all times, need to "approximately know" sorted order (?)
- usually study worst-case behavior for affine/pseudo-alg. data with no updates
- here: $\Theta(n^2)$
  - Ω: 
  - O: each pair passes $\leq$ once for affine — $O(1)$ for pseudo-alg.

Kinetic heap: [de Fonseca & de Figueiredo — IPL 2003]

- want find-min (& delete-min) in $O(\lg n)$
- could use kinetic predecessor ~ can do better
- store a min-heap
- certificates:



$x \le y$
$x \le z$

- event($x \le y$):
  - swap x & y in tree
  - update adjacent certificates

1. responsive: $O(\lg n)$
2. local: $O(1)$
3. compact: $O(n)$
4. efficient: $O(\lg n)$
   - $\Theta(n)$ changes to min in worst case
   - Ω:

   - O: once min changes $x \to y$, x cannot be min again

- claim $O(n \lg n)$ events in DS

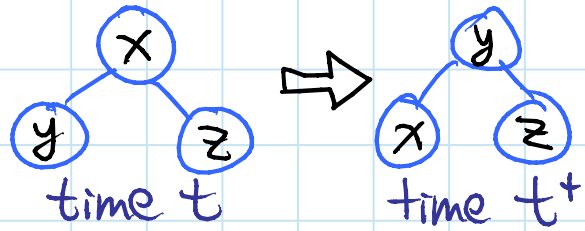
Proof:

- $\Phi(t) = \#$ events in future $> t$

$$= \sum_x \underbrace{\left(\begin{array}{c}\#\text{ descendants of } x \text{ @ time } t \text{ that} \\ \text{will overtake } x \text{ in future} > t\end{array}\right)}_{\Phi(t,x)}$$

- $\Phi(t,x) = \sum_{\substack{y \text{ child} \\ \text{of } x}} \underbrace{\left(\begin{array}{c}\#\text{ descendants of } y \text{ @ time } t \\ \text{that will overtake } x \text{ in} > t\end{array}\right)}_{\Phi(t,x,y)}$

- consider event at time t:



- $\Phi(t^+, v) = \Phi(t, v) \quad \forall v \neq x, y$

  (v gains/loses no descendants & isn't overtaken)

- $\Phi(t^+, x) = \underbrace{\Phi(t, x, y)}_{\text{remaining descends.}} - \underbrace{1}_{y}$

- $\Phi(t^+, y) = \Phi(t, y) + \Phi(t, y, z)$ (not x)

$$\leq \Phi(t, y) + \Phi(t, x, z)$$

(overtake $y \Rightarrow$ overtake x)

$$= \Phi(t, y) + \Phi(t, x) - \Phi(t, x, y)$$

$\Rightarrow \Phi(t^+) \leq \Phi(t) - 1$

- $\Phi(0) \leq \sum_x \underbrace{\#\text{ descendants of } x}_{O(\lg n)}$

$$= O(n \lg n) \qquad \square$$

Kinetic survey: [Guibas – DS Handbook 2005]

- 2D convex hull [Basch, Guibas, Hershberger 1999]
  - also diameter, width, min. area/perim. rectangle
  - efficiency = $O(n^{2+\varepsilon})/\Omega(n^2)$
  - OPEN: 3D?
- smallest enclosing disk: $O(n^3)$ events
  - OPEN: $O(n^{2+\varepsilon})$?
- $(1+\varepsilon)$-approximate diameter, smallest disk/rectangle in $(1/\varepsilon)^{O(1)}$ events [Agarwal & Har-Peled – SODA 2001]
- Delaunay triangulation [Albers, Guibas, Mitchell, Roos – IJCGA 1998]
  - $O(1)$ efficiency
  - OPEN: how many changes? $O(n^3)$ & $\Omega(n^2)$
- any triangulation:
  - $\Omega(n^2)$ changes even with Steiner points [Agarwal, Basch, de Berg, Guibas, Hershberger – SoCG 1999]
  - $O(n^{2+1/3})$ events [Agarwal, Basch, Guibas, Hershberger, Zhang – WAFR 2000]
  - OPEN: $O(n^2)$?
  - $O(n^2)$ events for pseudotriangulations 
- collision detection [Kirkpatrick, Snoeyink, Speckmann 2000] [Agarwal, Basch, Guibas, Hershberger, Zhang 2000] [Guibas, Xie, Zhang 2001] ← 3D
- MST
  - $O(m^2)$ easy: OPEN: $o(m^2)$? (sorted order of edge weights)
  - $O(n^{2-1/6})$ for H-minor-free graphs (e.g. planar) [Agarwal, Eppstein, Guibas, Henzinger – FOCS 1998]