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 \( a + b t \)
- initial velocity
- bounded-degree algebraic \( a + b t + ct^2 + \ldots \)
- pseudo-algebraic: any certificate of interest flips true/false \( \mathcal{O}(1) \) 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 $O(\log n)$
- let's try a BST
- certificates: $\exists x_i \leq x_{i+1}$
  where $x_1, x_2, \ldots, x_n$ is an in-order traversal
- failure_i = inf $\{ t \geq \text{now} | x_i(t) \geq x_{i+1}(t) \}$
  (next time certificate i will fail)
- advance(t):
  - while $t > Q.\min$:
    - now = $Q.\min$
    - event($Q.\ delete-min$)
  - now = t
- event ($x_i \leq x_{i+1}$): (in fact, $x_i = x_{i+1}$ now)
  - swap $x_i$ & $x_{i+1}$ in BST
  - add certificate $x_i \leq x_{i+1}'$
  - replace certificate $x_{i-1} \leq x_i$ with $x_{i-1} \leq x_i'$
  & certificate $x_{i+1} \leq x_{i+2}$ with $x_{i+1} \leq x_{i+2}$
  - update failure times in priority queue
Metrics:
1. **Responsive**: when certificate expires (event), can fix DS quickly \(O(\lg n)\)
2. **Local**: no object participates in many certs. \(\Rightarrow\) modify is fast \(O(1)\)
3. **Compact**: # certs is small \(\Rightarrow\) low space \(O(n)\)
4. **Efficient**: worst-case # DS events is small \(O(1)\)

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: \(O(n^2)\)

\(- \Omega: \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \rightarrow \quad \ldots \quad \)

\(- O: \) 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:
    \[
    \begin{array}{c}
    x \\
    y \\
    z \\
    \end{array}
    \]
    \[x \leq y \quad x \leq z\]
  - event ($x \leq y$):
    - 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
- $\Omega$: $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow \ldots$ etc.
- $O$: once min changes $x \Rightarrow y$, $x$ cannot be min again

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

- \( \Phi(t) = \# \text{ events in future } > t \)
  
  \[ = \sum_x \left( \# \text{ descendants of } x \text{ that will overtake } x \text{ in future } > t \right) \]

- \( \Phi(t, x) = \sum_{y \text{ of } x} \left( \# \text{ descendants of } y \text{ at time } t \right) \)
  
  \[ \text{(that will overtake } x \text{ in } > t) \]

- \( \Phi(t, x, y) \)

- Consider event at time \( t \):

  - \( \Phi(t^+, v) = \Phi(t, v) \) \( \forall v \neq x, y \)
    
    \( (v \text{ gains/loses no descendants} \) & \text{isn't overtaken} \)

  - \( \Phi(t^+, x) = \Phi(t, x, y) - 1 \)
    
    \( \text{remaining descendants of } y \)

  - \( \Phi(t^+, y) = \Phi(t, y) + \Phi(t, y, z) \) \( \text{(not } x) \)
    
    \[ \leq \Phi(t, y) + \Phi(t, x, z) \]
    
    \( \text{(overtake } y \Rightarrow \text{overtake } x) \)

  \[ = \Phi(t, y) + \Phi(t, x) - \Phi(t, x, y) \]

  \[ \Rightarrow \Phi(t^+) \leq \Phi(t) - 1 \]

- \( \Phi(0) \leq \sum_x \# \text{ descendants of } x \)
  
  \[ = O(n \log n) \]

\( \square \)
Kinetic survey:  
- 2D convex hull  \[ \text{[Basch, Guibas, Hershberger 1999]} \]
  - also diameter, width, min. area/perim. rectangle
  - efficiency = \( O(n^{2+\varepsilon}) / \Omega(n^2) \)
  - \text{OPEN}: 3D?
- smallest enclosing disk: \( O(n^3) \) events
  - \text{OPEN}: \( O(n^{2+\varepsilon}) \)?
- \( (1+\varepsilon) \)-approximate diameter, smallest disk/rectangle in \((\frac{1}{\varepsilon})o(1)\) events \[ \text{[Agarwal & Har-Peled - SODA 2001]} \]
- Delaunay triangulation \[ \text{[Albers, Guibas, Mitchell, Roos - IJCGA 1998]} \]
  - \( O(1) \) efficiency
  - \text{OPEN}: how many changes? \( O(n^3) \& \Omega(n^2) \)

- any triangulation:
  - \( \Omega(n^2) \) changes even with Steiner points \[ \text{[Agarwal, Basch, de Berg, Guibas, Hershberger - SoCG 1999]} \]
  - \( O(n^{2+1/3}) \) events \[ \text{[Agarwal, Basch, Guibas, Hershberger, Zhang - WAFR 2000]} \]
  - \text{OPEN}: \( O(n^2) \)??
  - \( O(n^2) \) events for pseudo triangulations

- collision detection \[ \text{[Kirkpatrick, Snoeyink, Speckmann 2000]} \]
  \[ \text{[Agarwal, Basch, Guibas, Hershberger, Zhang 2000]} \]
  \[ \text{[Guibas, Xie, Zhang 2001]} \leq 3D \]

- MST 
  - \( O(m^2) \) easy: \text{OPEN}: \( o(m^2) \)?
  - \( O(n^{2-1/6}) \) for H-minor-free graphs (e.g. planar) \[ \text{[Agarwal, Eppstein, Guibas, Henzinger - FOCS 1998]} \]