Today: Dynamic Optimality II (of 2)
- lower bounds:
  - independent rectangles
  - Wilber 1 & 2
  - signed greedy
- Tango trees: $O(\log \log n)$-competitive

Recall:
- point set is a valid BST execution
  $\Leftrightarrow$ arborally satisfied set:
    rectangle spanned by two points
    not on a horizontal/vertical line
    contains another point
- Greedy algorithm conjectured $O(\text{optimal})$
- can be simulated online
Lower bounds: [Demaine, Harmon, Iacono, Kane, Patrascu]

Independent rectangles are unsatisfied & \( \Rightarrow \) input point set (accesses) no corner is strictly inside another

![Dependent and independent rectangles]

**Theorem:** \( \text{OPT} \geq |\text{input}| + \frac{1}{3} \max \# \text{ independent rectangles} \)

Signed rectangles: \( \square \) & \( \square \) types
- \( \square \)-satisfied if all \( \square \) rectangles have another pt.
- \( \text{OPT}_{\square} \) = smallest \( \square \)-satisfied superset of points

**Lemma:** \( \text{OPT}_{\square} \geq |\text{input}| + \max \# \text{ independent } \square \text{-rectangles} \)

**Proof:**
1. find rectangle in indep. set \\& vertical line hitting just it \\
   \( \Rightarrow \) segment with endpoints on top & bottom edges of rectangle
2. find horizontally adjacent pts. of \( \text{OPT}_{\square} \) in rect. crossing line
3. charge indep. rectangle to those points
Assume input x & y coords. all distinct

1: take the widest rectangle

- sharing-a rects. left of sharing-b's (indep)
- sharing-neithers fit in between vertical edges
  => room left for vertical line

2: take p = topmost rightmost point in rectangle & left of line (e.g. a)

q = bottommost leftmost point in rectangle & right of line & not below p (e.g. b)

3: p & q are not in any other common rectangle
  => pair won't get charged again
  - in any horizontal chain of charges
  <= 1 in input (by distinct y's)
  => added > # indep. rectangles
Wilber's second lower bound:
- given input (access) point set
- for each point \( p \):
  - look at orthogonally visible points below \( p \)
  - count # alternations between left/right of \( p \)
- sum over all \( p \)

Proof: independent rectangle \( \forall \) alternation:

**Conjecture:** \( \text{OPT} = \Theta(\text{Wilber}^2) \)

**Key-independent optimality:** [Iacono - ISAAC 2002]
- suppose key values are "meaningless"
  - might as well permute them uniformly at random
- claim: \( E[\text{OPT}] = \text{working-set bound} \)
  - splay trees are key-indep. optimal
- proof sketch: \( E[\text{Wilber}^2(x_i)] = \Theta(\lg t_i) \)
  (expected # changes to max. in random permutation)
Wilber's first lower bound: [Wilber - SICOMP 1989]
- fix a lower-bound tree \( P \) on same keys 
  (e.g. perfect binary tree)
- for each node \( y \) of \( P \):
  count # alternations in \( x_1, x_2, \ldots, x_n \)
  between accesses in left & right subtrees of \( y \) 
  (ignoring accesses to \( y \) or outside \( y \)'s subtree)
- sum over all \( y \)

Proof: independent rectangle alternation

Example: bit-reversal sequence

<table>
<thead>
<tr>
<th>000</th>
<th>010</th>
<th>011</th>
<th>100</th>
<th>101</th>
<th>110</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>6</td>
<td>3</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>4</td>
</tr>
</tbody>
</table>

\( \Rightarrow \) # alternations at \( y = \) size of \( y \)'s subtree
\( \Rightarrow \) Wilber 1 = \( \Theta(n \log n) \)
\( \Rightarrow \) OPT = \( \Theta(n \log n) \)

[OPEN]: Any access sequence \( \exists \) tree \( P \) such that 
OPT = \( \Theta(\text{Wilber 1}) \)
Tango trees: \cite{DemaineHarmonIaconaPatrascu2007}
- \(O(\lg \lg n)\)-competitive online BST
- \(P\) = perfect BST on \(n\) keys
  - define preferred child of node \(y\) in \(P\) to be
    - left if accessed left subtree of \(y\) more recently
    - right if accessed right subtree of \(y\) more recently
    - none if no access to either subtree yet
  - preferred path = chain of preferred child pointers
  - partition of nodes of \(P\)
- idea: store each preferred path in auxiliary tree
  - conceptually separate balanced BST (e.g. AVL)
  - leaves link to roots of aux. trees of children paths
  - has \(\leq \lg n\) nodes (height of perfect \(P\))
  - supports search in \(O(\lg \lg n)\) time
- search starts at top aux. tree (containing root of \(P\))
  - each jump to next aux. tree = nonpreferred edge
    - preferred edge change = \(+1\) in Wilber 1
  - \(k\) jumps \(\Rightarrow UB\ k\), \(UB\ (k+1)\cdot O(\lg \lg n)\)
  - \(O(\lg \lg n)\)-competitive... if we can update preferred edges OK
Auxiliary trees:
- changing a preferred child = cutting one path & joining two paths:
  - if aux. trees were sorted by depth, this would be like split & concatenate
  - depth >d translates to interval of keys
  \[ \Rightarrow \text{can implement cuts & joins with } O(1) \text{ splits & concatenates} \]
  - each costs \( O(lg (\text{aux. tree})) = O(lg \lg n) \)

In one tree: mark roots of aux. trees
- modify split & concat. to ignore children trees & manipulate adjacent trees:
Signed Greedy:
- sweep as in Greedy
- only satisfy \( \square \) boxes
- for every added point, get independent \( \square \)-rectangle
\[ \Rightarrow \text{get lower bound: } \square \text{-Greedy} \]

**Theorem:** \[
\max \{ \square \text{-Greedy}, \square \text{-Greedy} \} = \Theta(\text{biggest independent-rectangle LB})
\]

**Proof:** define \( \text{OPT}_{\square} = \text{smallest union of } \square \text{-satisfying superset} \cup \square \text{-satisfying superset} \)

\[
\text{OPT} \geq \text{OPT}_{\square} \\
\geq |\text{input}| + \frac{1}{2} \max \{ \text{independent rectangles} \} \\
\geq \frac{1}{2} \max \{ \square \text{-Greedy}, \square \text{-Greedy} \} \\
\geq \frac{1}{2} \max \{ \text{OPT}_{\square} \cup \text{OPT}_{\square} \} \\
\geq \frac{1}{4} (\text{OPT}_{\square} + \text{OPT}_{\square}) \\
\geq \frac{1}{4} \text{OPT}_{\square} \\
\Rightarrow \text{constant-factor sandwich} \]

**Summary:** so close!

**PROJECT:** compare UBs & LBs for many pt. sets