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

Recall:
- point set is a valid BST execution
  $\Longleftrightarrow$ 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** & \( \Rightarrow \) in input point set (accesses) no corner is strictly inside another

![Diagrams of dependent and independent rectangles]

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

**Signed rectangles:** \( [ ] \) & \( [ ] \) types
- \( [ ] \)-satisfied if all \( [ ] \) rectangles have another pt.
- \( OPT_{[ ]} \) = smallest \( [ ] \)-satisfied superset of points

**Lemma:** \( OPT_{[ ]} \geq \text{input} + \max \# \text{ independent } [ ] \)-rectangles

**Proof:**
1. find rectangle in indep. set & vertical line hitting just \( b \) segments with endpoints on top & bottom edges of rectangle
2. find horizontally adjacent pts. of \( OPT_{[ ]} \) in rect. crossing line \( a \)
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 \( \& \) alternation:

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

Key-independent optimality:

- 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: \( E[\text{Wilber}^2(x_i)] = \Theta(\log t_i) \)
  (expected \# changes to max. in random permutation)
Wilber's first lower bound: \[\text{[Wilber - SICOMP 1989]}\]
- 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

\[
\begin{align*}
000 & \quad 0 \\
001 & \quad 4 \\
010 & \quad 2 \\
011 & \quad 6 \\
100 & \quad 1 \\
101 & \quad 5 \\
110 & \quad 3 \\
111 & \quad 7 \\
\end{align*}
\]

\(\Rightarrow\) # alternations at \(y\) = size of \(y\)'s subtree

\(\Rightarrow\) Wilber 1 = \(\Theta(n \log n)\)

\(\Rightarrow\) OPT = \(\Theta(n \log n)\)

OPEN: \(\exists\) access sequence \(\exists\) tree \(P\) such that

\(\text{OPT} = \Theta(\text{Wilber 1})\)
Tango trees: [Demaine, Harmon, Iacono, Patrascu - SICOMP 2007]
- \(O(\log \log n)\)-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 \(\leq \log n\) nodes (height of perfect \(P\))
  \(\Rightarrow\) supports search in \(O(\log \log n)\) 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 \(\Rightarrow UB\) \(k\), \(ub\) \((k+1)\cdot O(\log \log n)\)
  \(\Rightarrow O(\log \log n)\)-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
  ⇒ can implement cuts & joins with \(O(1)\) 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: \[ \max \left\{ \square \text{-Greedy, } \Box \text{-Greedy} \right\} = \Theta(\text{biggest independent-rectangle LB}) \]
Proof: define \( \text{OPT} \square \) = smallest union of \( \square \)-satisfying superset & \( \Box \)-satisfying superset

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

Summary: so close!

Greedy \( \square \& \Box \) UB vs. Signed Greedy \( \square + \Box \) LB

PROJECT: compare UBs & LBs for many pt. sets