Lower bounds: [Demaine, Harmon, Iacono, Kane, Pătraşcu]

Independent rectangles are unsatisfied & no corner is strictly inside another.

Theorem: \( \text{OPT} \geq \text{input} + \frac{1}{2} \max \# \text{ independent rectangles} \)

Signed rectangles: \( \square \) & \( \bigsymbol{\square} \) types

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

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

Proof: ① find rectangle in indep. set & vertical line hitting just it

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-neither 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 ⇒ won't get charged again
- at most one is in input; other is added □
Wilber's second lower bound:
- 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 $\triangledown$ 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: $E[\text{OPT}]$ = working-set bound
  ⇒ splay trees, Greedy 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: [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 \( y \)

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: Any access sequence \( \exists \) tree \( P \) such that
\( OPT = \Theta(Wilber\ 1) \)
Tango trees: [Demaine, Harmon, Iacono, Patrascu - SICOMP 2007]
- O(lg lg n)-competitive online BST
- 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 ≤ 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 ⇒ UB k, UB (k+1)·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 \(O(1)\) 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 \( \Box \) boxes
- for every added point, get independent \( \Box \)-rectangle
\( \Rightarrow \) get lower bound: \( \Box \)-Greedy

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

**Proof:** define \( \text{OPT}^{\Box} = \text{smallest union of } \Box \text{-satisfying superset} \)

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

**Summary:** so close!

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

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