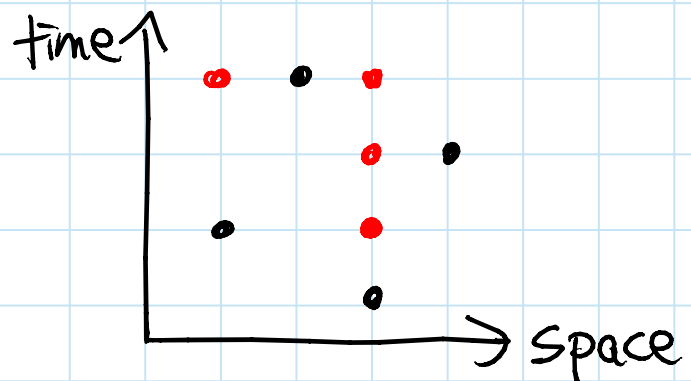


TODAY: Dynamic Optimality II (of 2)

- lower bounds:
 - independent rectangles
 - Wilber 1 & 2
 - signed greedy
- Tango trees: $O(\lg \lg n)$ -competitive

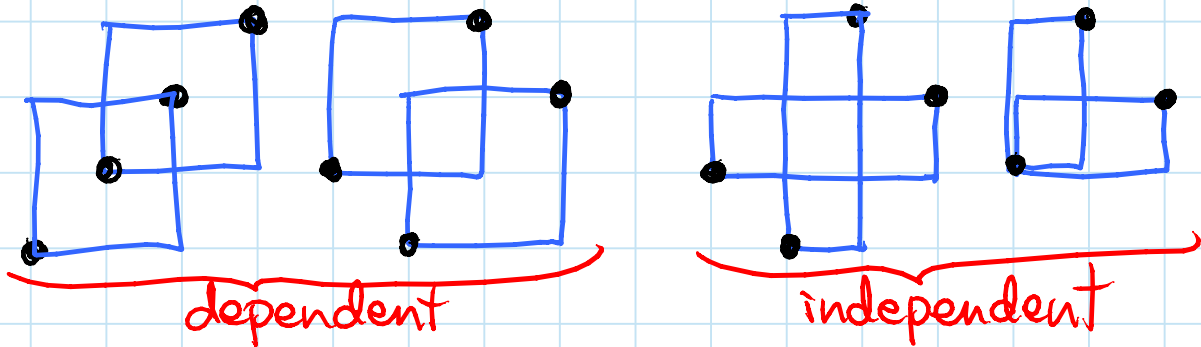
Recall:

- point set is a valid BST execution
- \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]

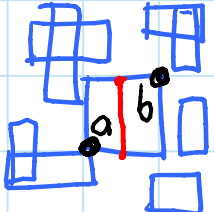

Independent rectangles are unsatisfied &
↳ in input point set (accesses)
no corner is strictly inside another



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

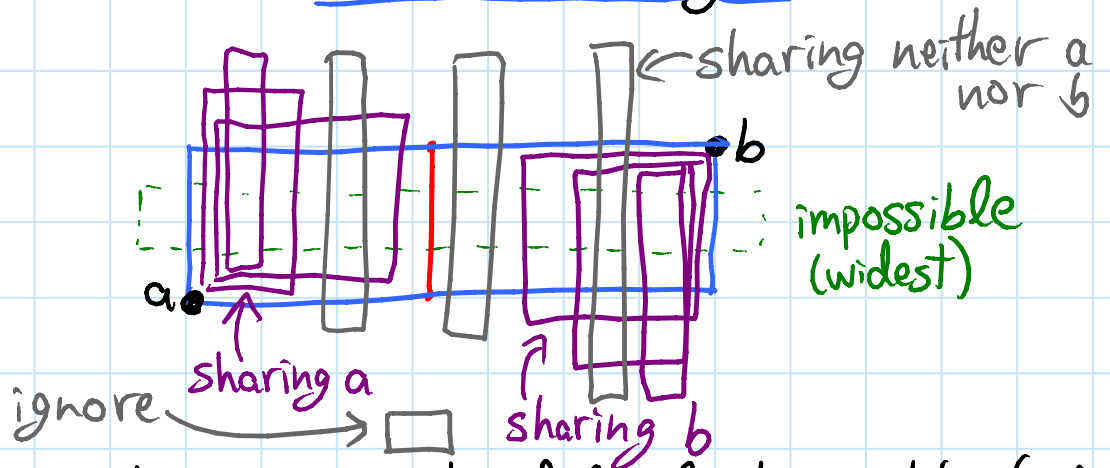
Signed rectangles:  types
- \square -satisfied if all \square rectangles have another pt.
- $OPT_{\square} =$ smallest \square -satisfied superset of points

Lemma: $OPT_{\square} \geq |input| + \max \# \text{ independent } \square\text{-rectangles}$

- Proof:
- ① find rectangle in indep. set & vertical line hitting just it
↳ segments with endpoints on top & bottom edges of rectangle 
 - ② find horizontally adjacent pts. of OPT_{\square} in rect. crossing line 
 - ③ charge indep. rectangle to those points

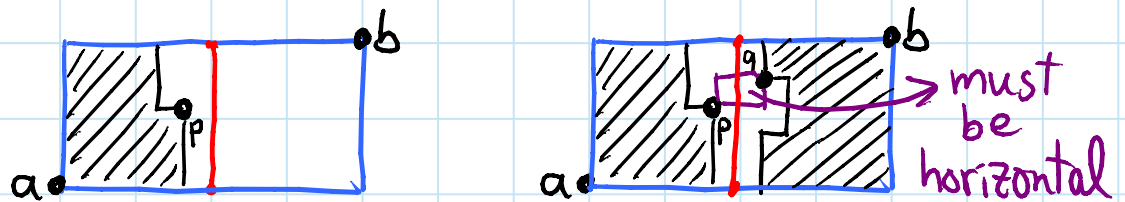
Assume input x & y coords. all distinct $\vdots \Rightarrow \vdots$

①: take the widest rectangle



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

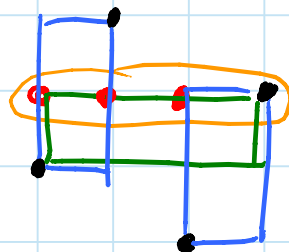
②: take p = topmost rightmost point in rectangle & left of line (e.g. a)
 q = bottommost leftmost point in rectangle & right of line & not below p (e.g. b)



③: p & q are not in any other common rectangle
 \Rightarrow pair won't get charged again

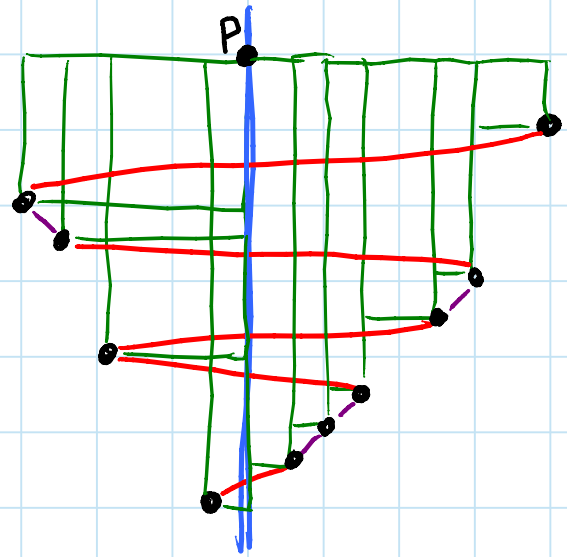
- in any horizontal chain of charges

≤ 1 in input (by distinct y 's)
 \Rightarrow added \geq # indep. rectangles



Wilber's second lower bound: [Wilber - SICOMP 1989]

- 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: 

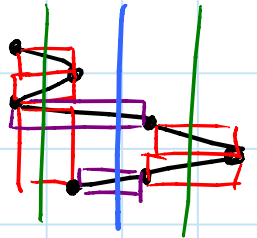
Conjecture: $OPT = \Theta(\text{Wilber } 2)$

Key-independent optimality: [Iacono - ISAAC 2002]

- suppose key values are "meaningless"
- \Rightarrow might as well permute them uniformly at random
- claim: $E[OPT] = \text{working-set bound}$
 - \Rightarrow 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]

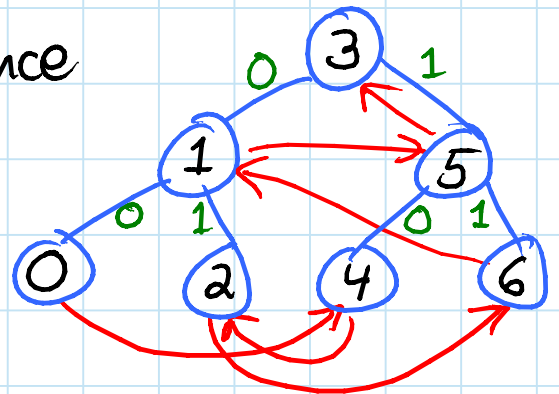
- fix a lower-bound tree P on same keys
(e.g. perfect binary tree)
- for each node y of P :
count # alternations in x_1, x_2, \dots, x_n
between accesses in left & right subtrees of y
(ignoring accesses to y or outside y 's subtree)
- sum over all y



Proof: independent rectangle ∇ alternation

Example: bit-reversal sequence

000	0
001	4
010	2
011	6
100	1
101	5
110	3
111	7

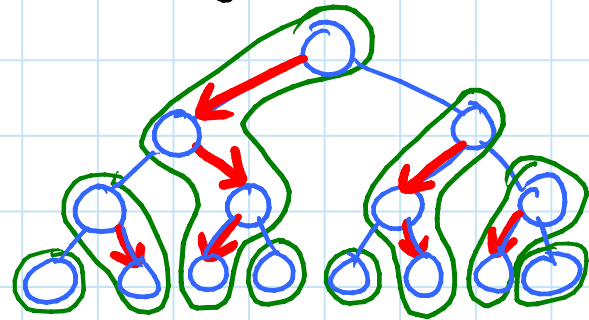


- \Rightarrow # alternations at y = size of y 's subtree
- \Rightarrow Wilber 1 = $\Theta(n \lg n)$
- \Rightarrow OPT = $\Theta(n \lg n)$

OPEN: \forall access sequence \exists tree P such that
OPT = Θ (Wilber 1)

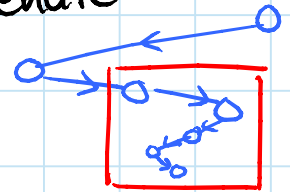
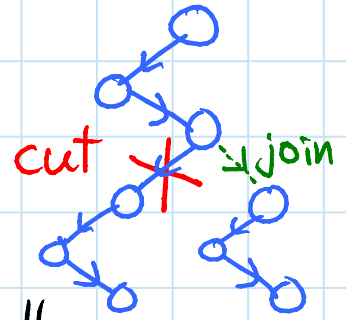
Tango trees: [Demaine, Harmon, Iacono, Patrascu - SICOMP 2007]

- $O(\lg \lg n)$ -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)
 - \Rightarrow 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 LB k , UB $(k+1) \cdot O(\lg \lg n)$
 - $\Rightarrow O(\lg \lg n)$ -competitive ... if we can update preferred edges OK



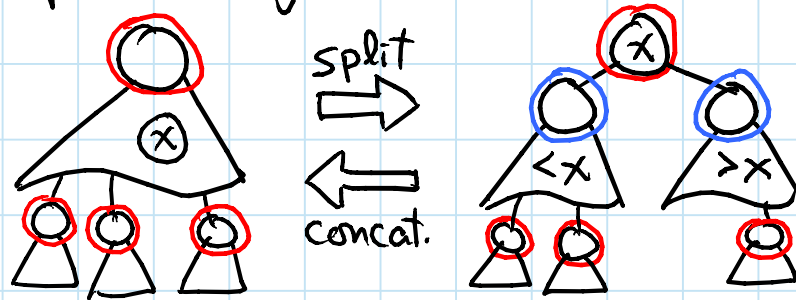
Auxiliary trees:

- 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
- ⇒ can implement cuts & joins with $O(1)$ splits & concatenates
- each costs $O(\lg(\text{aux. tree})) = O(\lg \lg n)$



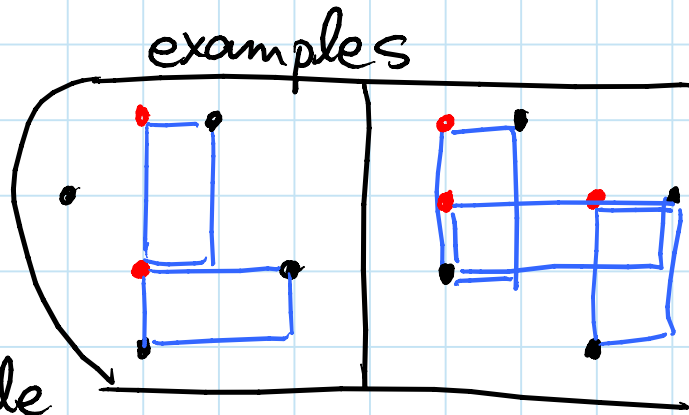
In one tree: mark roots of aux. trees

- 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 get lower bound: \square -Greedy



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

Proof: define $\text{OPT}_{\square} = \text{smallest union of } \square\text{-satisfying superset \& } \square\text{-satisfying superset}$

$$(\text{OPT} \geq) \text{OPT}_{\square}$$

$$\geq |\text{input}| + \frac{1}{2} \max. \# \text{ independent rectangles}$$

$$\geq \frac{1}{2} \max\{\square\text{-Greedy}, \square\text{-Greedy}\}$$

$$\geq \frac{1}{2} \max\{\text{OPT}_{\square}, \text{OPT}_{\square}\}$$

$$\geq \frac{1}{4} (\text{OPT}_{\square} + \text{OPT}_{\square})$$

$$\geq \frac{1}{4} \text{OPT}_{\square}$$

\Rightarrow constant-factor sandwich \square

Summary: so close!

Greedy
 \square & \square
UB

vs.

Signed Greedy
 \square + \square
LB

PROJECT: compare UBs & LBs for many pt. sets

MIT OpenCourseWare
<http://ocw.mit.edu>

6.851 Advanced Data Structures
Spring 2012

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.