Settling the relationship between Wilber's bounds for dynamic optimality

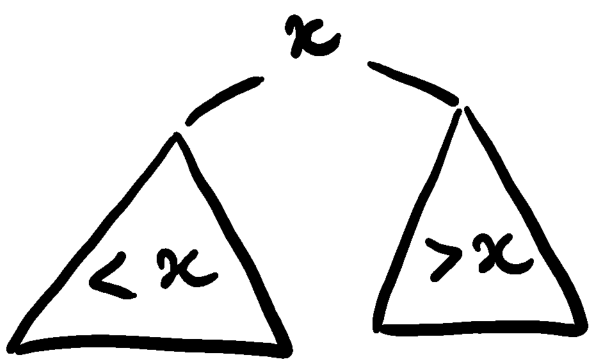
Victor Lecomte   Omri Weinstein

1

Binary Search Trees and the Dynamic Optimality conjecture

Binary Search Trees

One simple rule



$\Rightarrow$ easy to find keys

- start at root
- if too big, go left
- if too small, go right

Good for (static) dictionaries:

if balanced, log n time per query

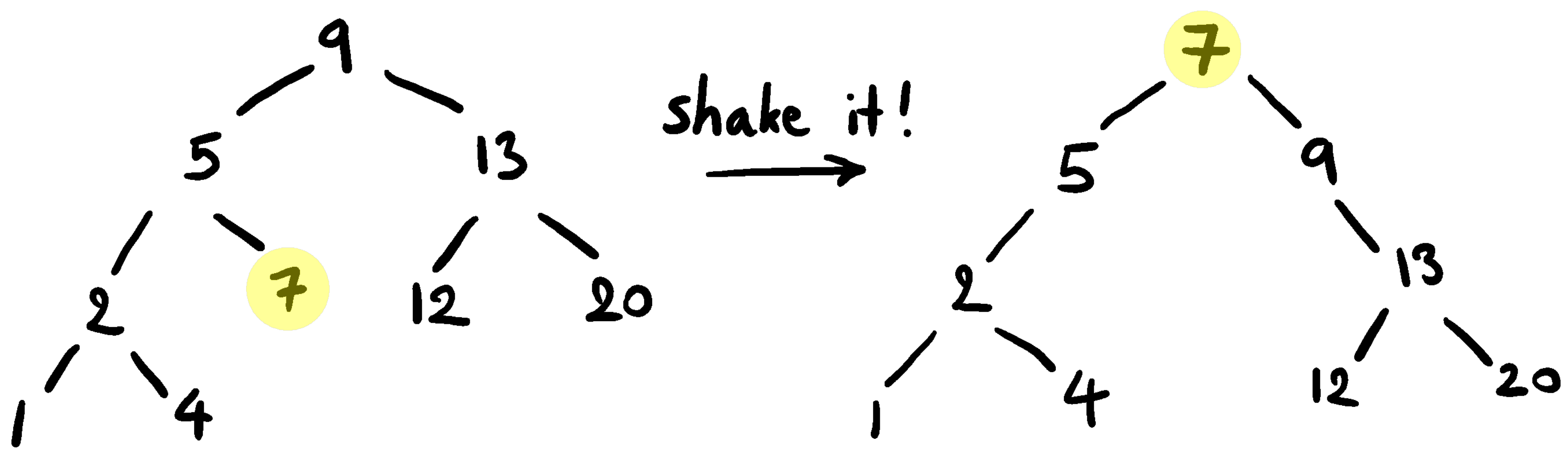
Can we do better?

In the worst case, no.

But what if one key is queried often?

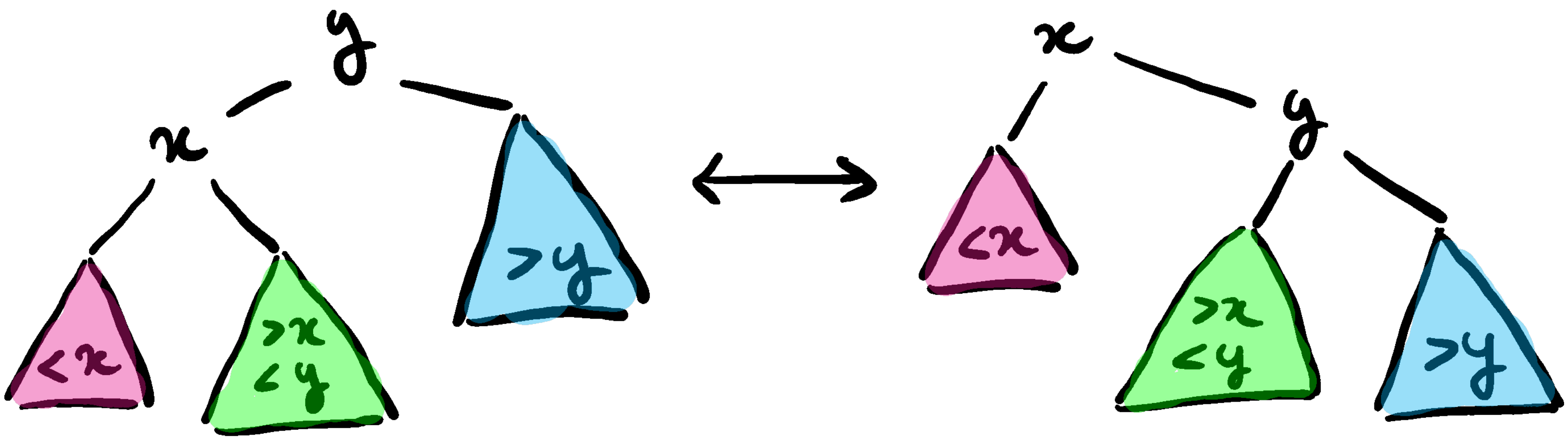
Would be nice to put it near the root.

Want to adapt.



How to tweak your BST

Rotations!



✓ VALID preserve BST property

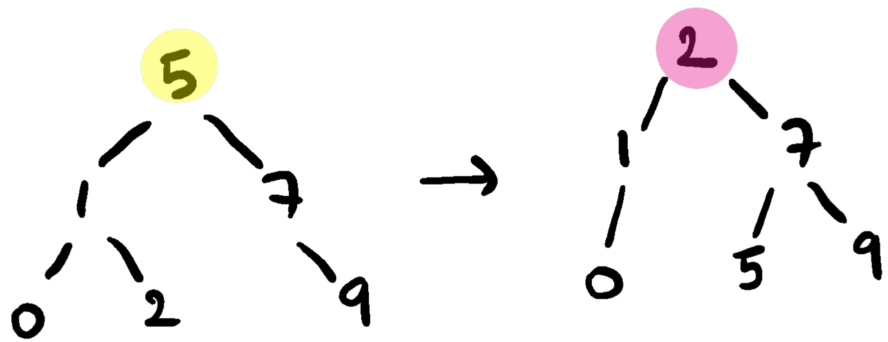
✓ POWERFUL can transform into any other BST on the same nodes

Rotations can help in many cases

"time locality"

$X = 5, 5, 5, 5, 2, 2, 2, \ldots$

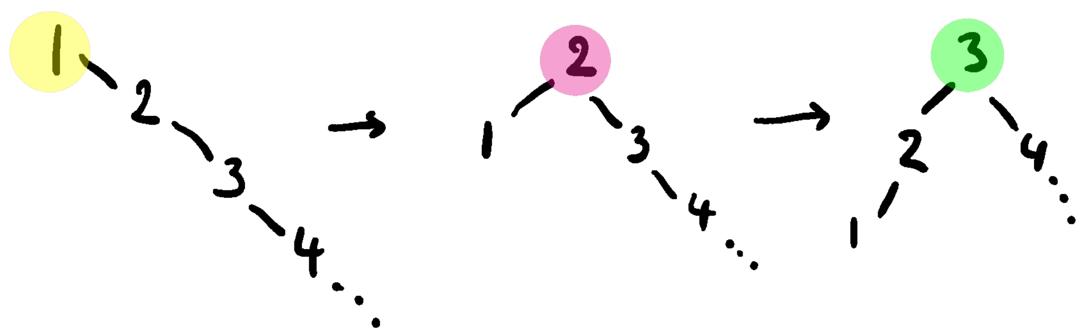
move to root



"space locality"

$X = 1, 2, 3, 5, 4, 6, 8, 9, \ldots$

"pull on the rope"



… or mixes of the two, and much more!

Can we get everything?

Is there one single algorithm that does almost as well as all others?

NOTE: can't use best strategy in hindsight, need to adapt on the fly

Too much to ask for?

Our benchmark

Let $X \in \underbrace{[n]}_{\text{keys}}{}^m$ $(m \geq n)$ be a sequence of queries.

To "execute" X,

- start with a fixed initial BST on $[n]$
  (e.g. 1 \ 2 . . . n , don't really care)
- when receive query x_i
  - → start pointer at root
  - → can move up, down, rotate
  - → access node x_i along the way

each costs 1

$\mathrm{OPT}(X)$:= the lowest cost of executing X

Properties of OPT(x)

a. Between $\Theta(m)$ and $\Theta(m \log n)$

e.g. $X = 1, 2, \dots, n$

by counting argument

b. Don't care about constant factors

(definition-sensitive)

c. Don't care about $\pm O(m)$

(from a. and b.)

"Dynamic Optimality"

$\exists$? an online BST algorithm A s.t.

doesn't know X in advance

$\forall X \quad \text{cost}_A(X) = O(\text{OPT}(X))$

instance optimality

operations A takes to execute X

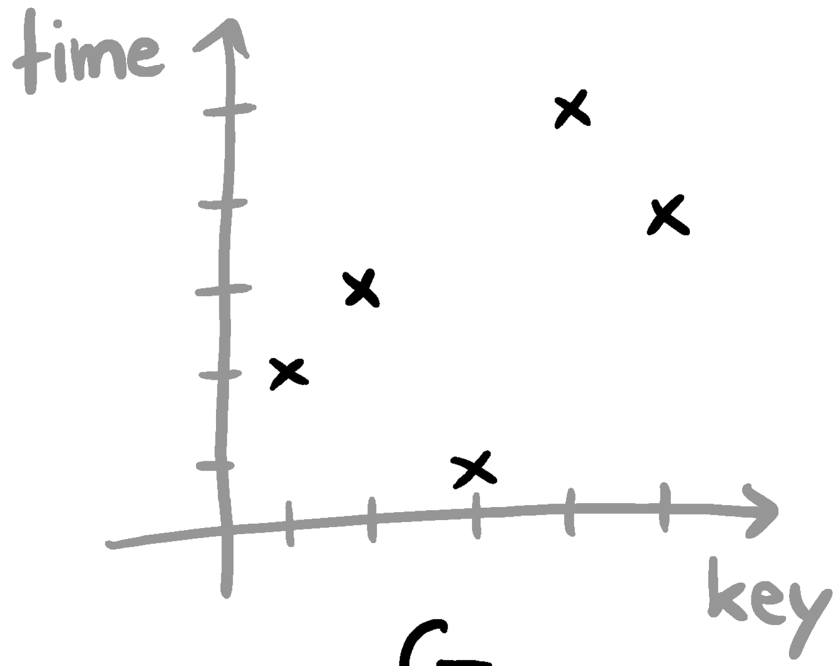
CANDIDATES
- [ST'85] Splay Trees
- [Lucas'88], [Fox'11] Greedy

But no idea how to prove it.

The geometric view

$G_X :=$ "plot" of X over time

$3, 1, 2, 5, 4 \longrightarrow$

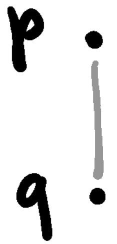


X (sequence)

G_X (set of points)

"Satisfied" rectangles

- A pair of points (p, q) is satisfied if

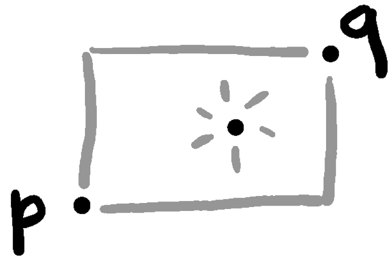


vertically aligned

or

horizontally aligned

or



their (closed) bounding box contains another point

- A set of points P is satisfied if

$\forall\, p, q \in P$, (p, q) is satisfied

From BSTs to the plane

Theorem [DHIKP'09]

$$OPT(X) = \Theta\left(\min_{\substack{Y \supseteq G_X \\ Y \text{ satisfied}}} |Y| \right)$$

INTUITION

- i^{th} row of $Y \simeq$ nodes visited at time i
- exercise: show $\boxed{\geq}$

2

Lower bounds on OPT(X)

OPT(X) is annoying

All known definitions are like

- minimum over all algorithms
- minimum over all supersets of G_X

$\Rightarrow$ to prove an algorithm matches $(\le)$ OPT,

need to consider all $\left\{ \begin{array}{l} \text{other algos} \\ \text{sets of points} \end{array} \right.$

How about a "closed form"?

Wilber's bounds

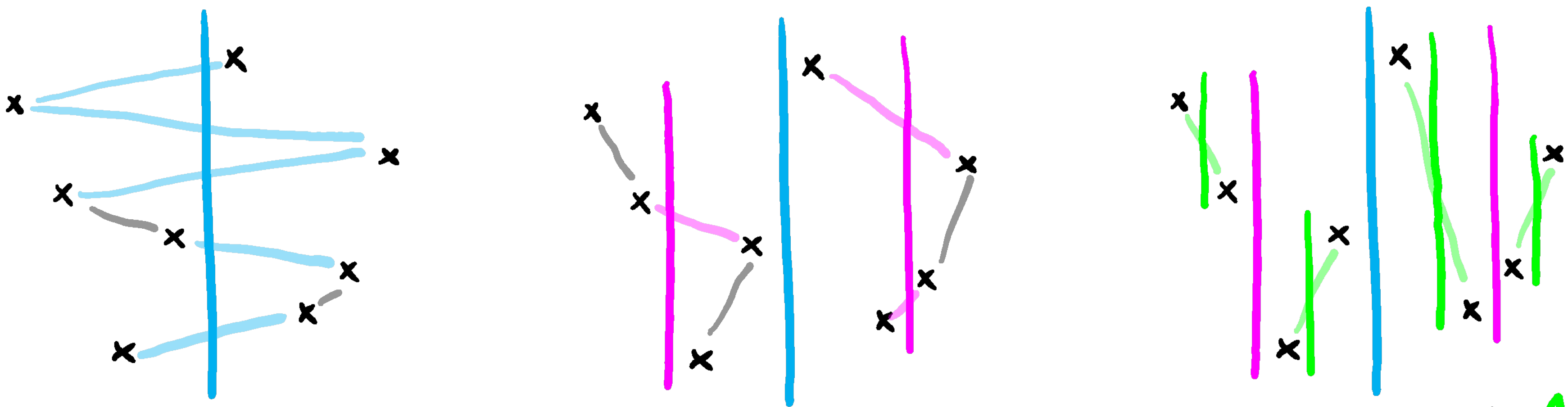
[Wilber'89]: 2 lower bounds on $OPT(x)$

1) the Alternation bound $Alt(x)$
2) the Funnel bound $Funnel(x)$

As far as we knew, $Alt(x) = Funnel(x) = OPT(x)$!

The Alternation bound

- Split G_X vertically
- Count how many times the points switch sides as you go up
- Recurse on either side



$$5 + 1 + 2 + 1 + 1 + 1 + 1 = 12$$

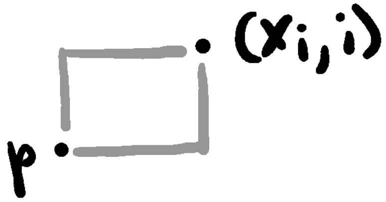
$\mathrm{Alt}(X) :=$ max number of side switches you can get overall

(if you split in the optimal order)

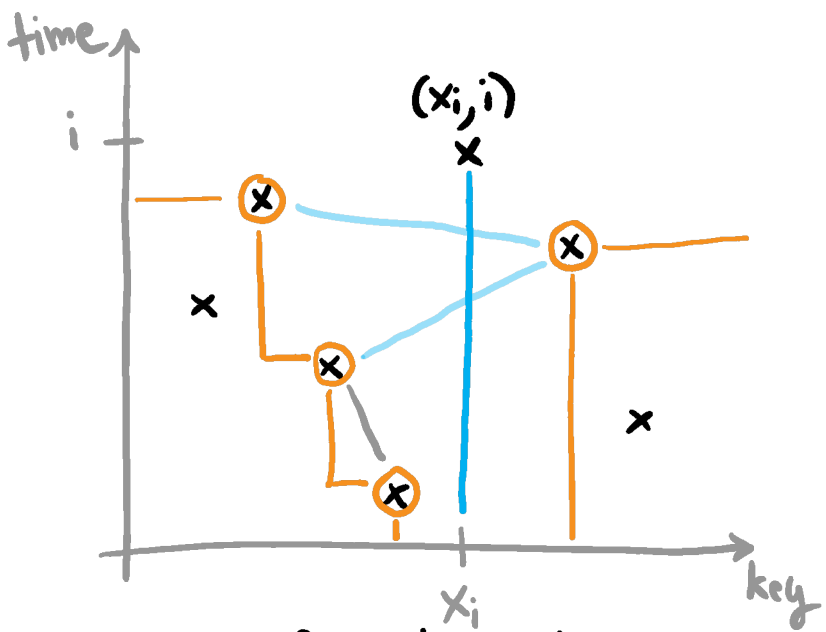
The Funnel bound

Fix time $i \in [m]$.

Consider the points $p \in G_x$ before time i s.t.



empty ("funnel points").



O = funnel point

$\text{Funnel}(x,i) :=$ # times the funnel points switch sides across $x = x_i$ as you go up (here, 2)

$$\text{Funnel}(x) := \sum_{i=1}^{m} \text{Funnel}(x,i)$$

Our results

Popular belief about Wilber's bounds:

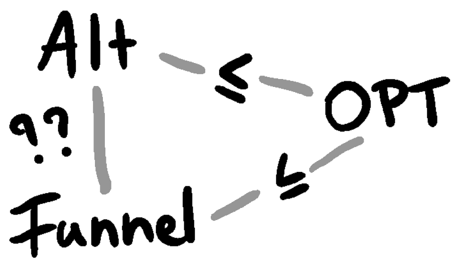
a) Alt not tight ($\ll \mathrm{OPT}(x)$ for some x)

b) Alt "weaker than" Funnel

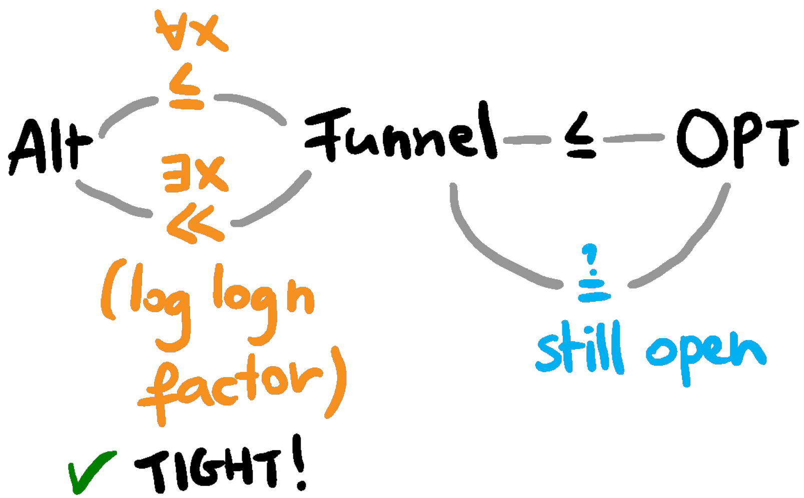
c) Funnel tight ($= \Theta(\mathrm{OPT})$)

we prove a) + b)

BEFORE



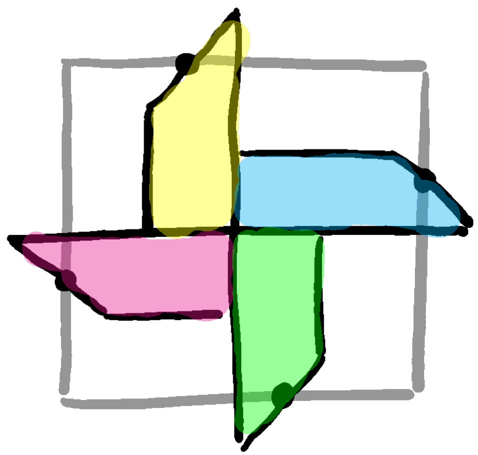
AFTER



+ BONUS : nicer definition for Funnel

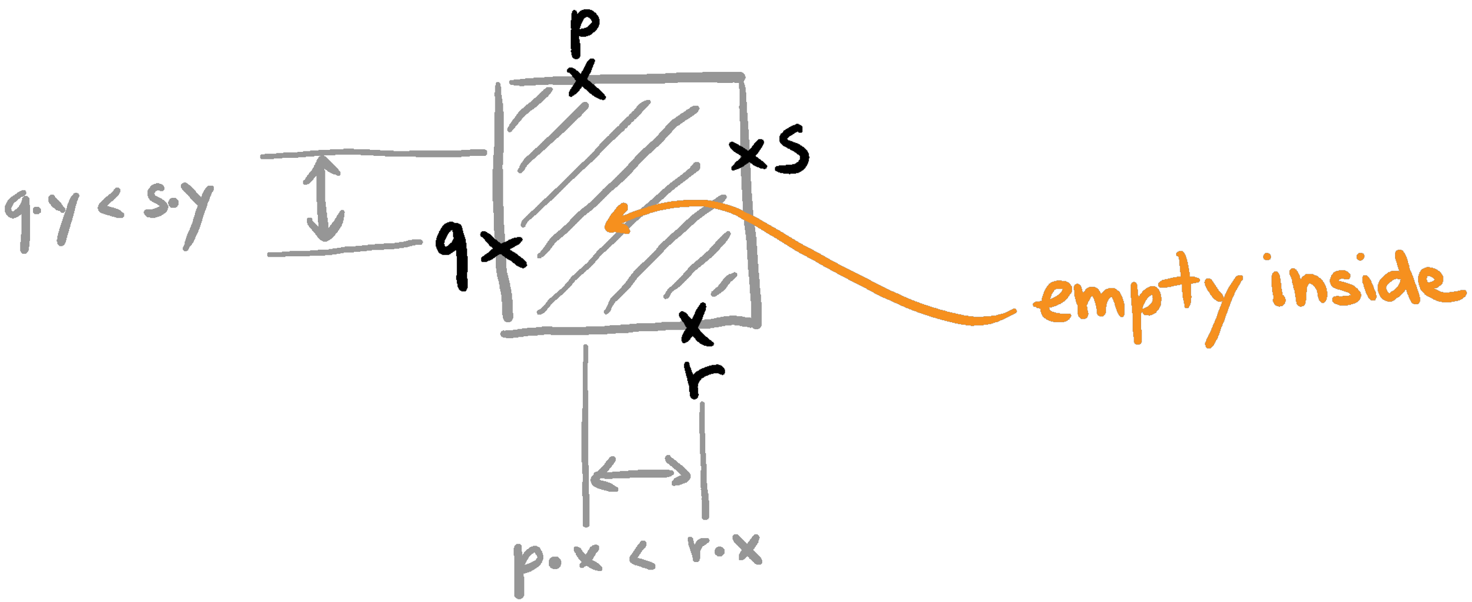
3

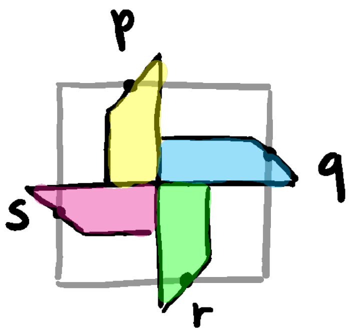
Counting pinwheels



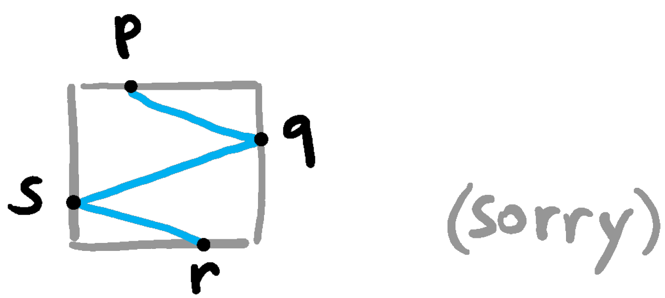
Pinwheels??

A pinwheel is a quadruple (p, q, r, s) of points in this configuration:





looks like a pinwheel



called z-rectangle in the paper

"Pinwheel bound"

Let $\text{Pinwheel}(x) := \#\text{pinwheels in } G_x$.

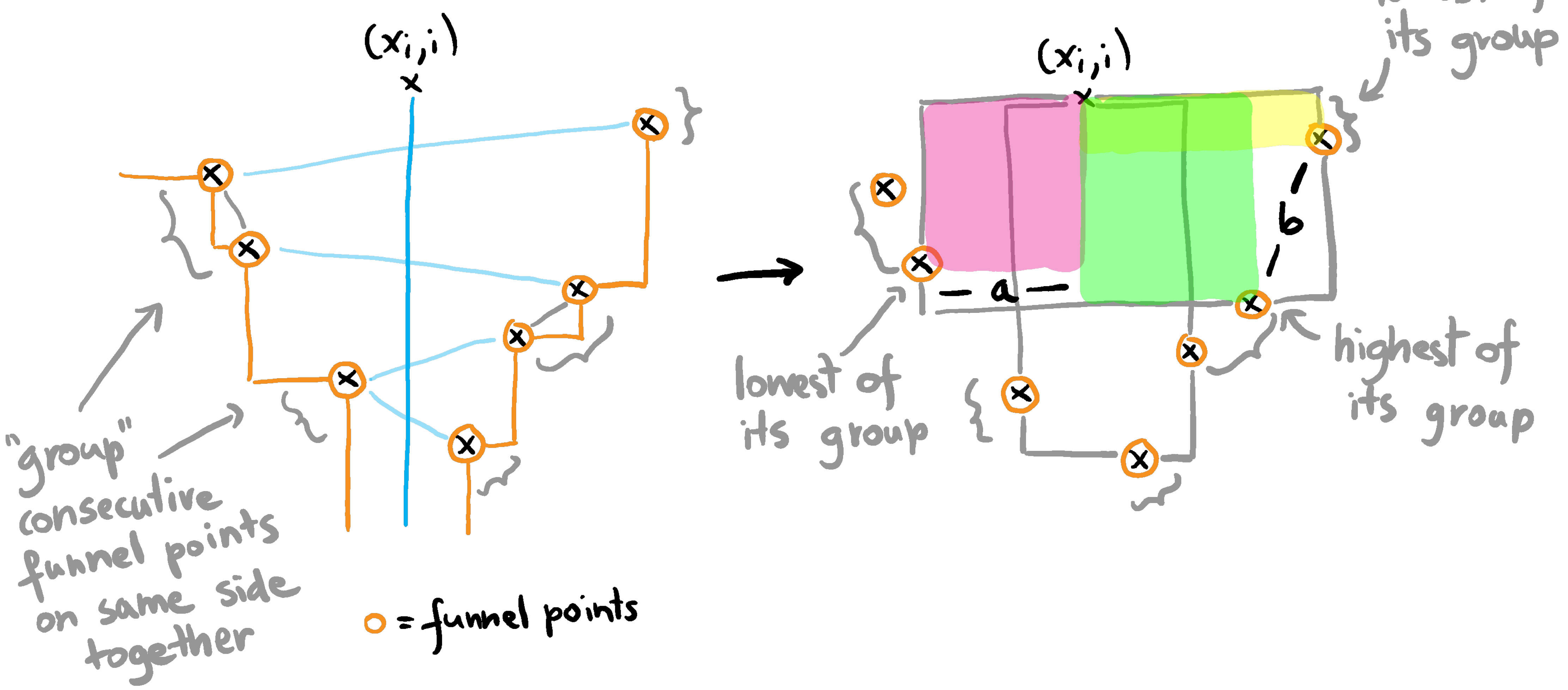
<u>Theorem</u>

$$\text{Funnel}(x) = \Theta(\text{Pinwheel}(x)) \pm O(m)$$

who cares

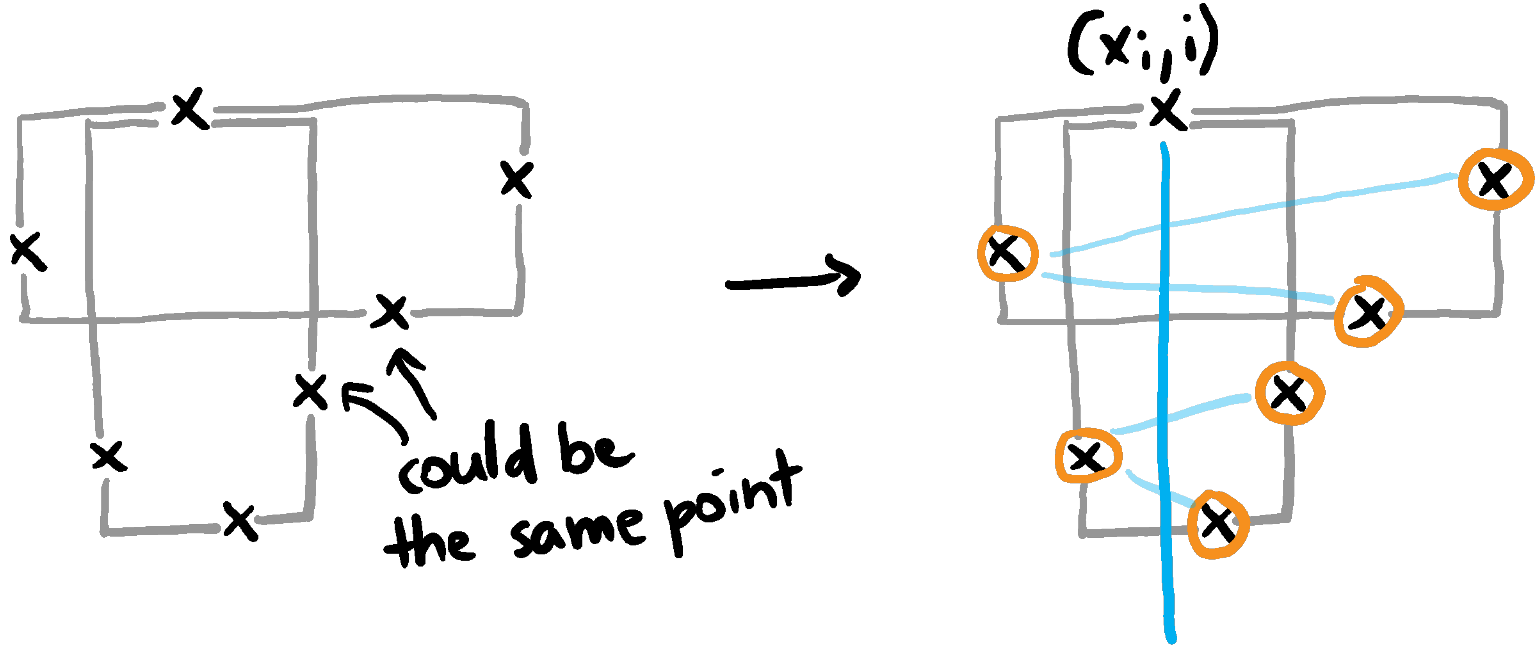
Pinwheel ≥ Funnel

The bigger Funnel(x,i), the more pinwheels with (x_i,i) on top.



Funnel $\geq$ Pinwheel

When several pinwheels have the same top point, they must be positioned this way relative to each other:



implies side switches

$\Rightarrow$ contributes to Funnel(x, i)

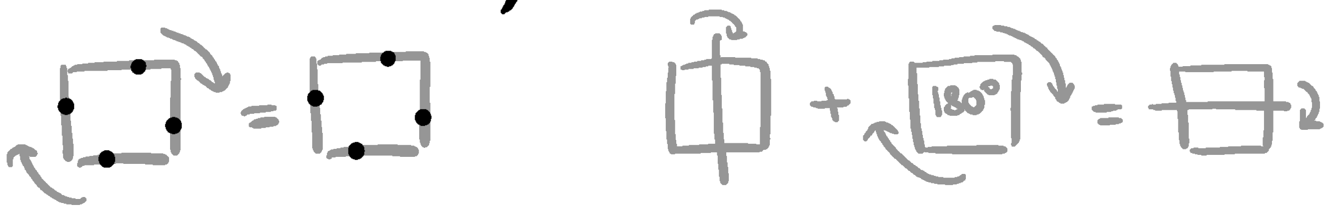
4

What now?

Consequences for Funnel

- much nicer definition
  (just count pinwheels)

- invariant to rotations, vertical flips



good news, because OPT too!

Open: OPT $\stackrel{?}{=}$ Funnel

Reminder:

$\forall x$

$\text{Alt} \leq \text{Funnel} \leq \text{OPT}$

$\exists x$

$\text{Alt} \ll \text{Funnel}$

(log log n factor)

$\text{Funnel} \stackrel{?}{=} \text{OPT}$

still open

If true

- very neat
- very useful for comparing algorithms to OPT

Thanks!

To learn more:

"In pursuit of the dynamic optimality conjecture"

survey by John Iacono