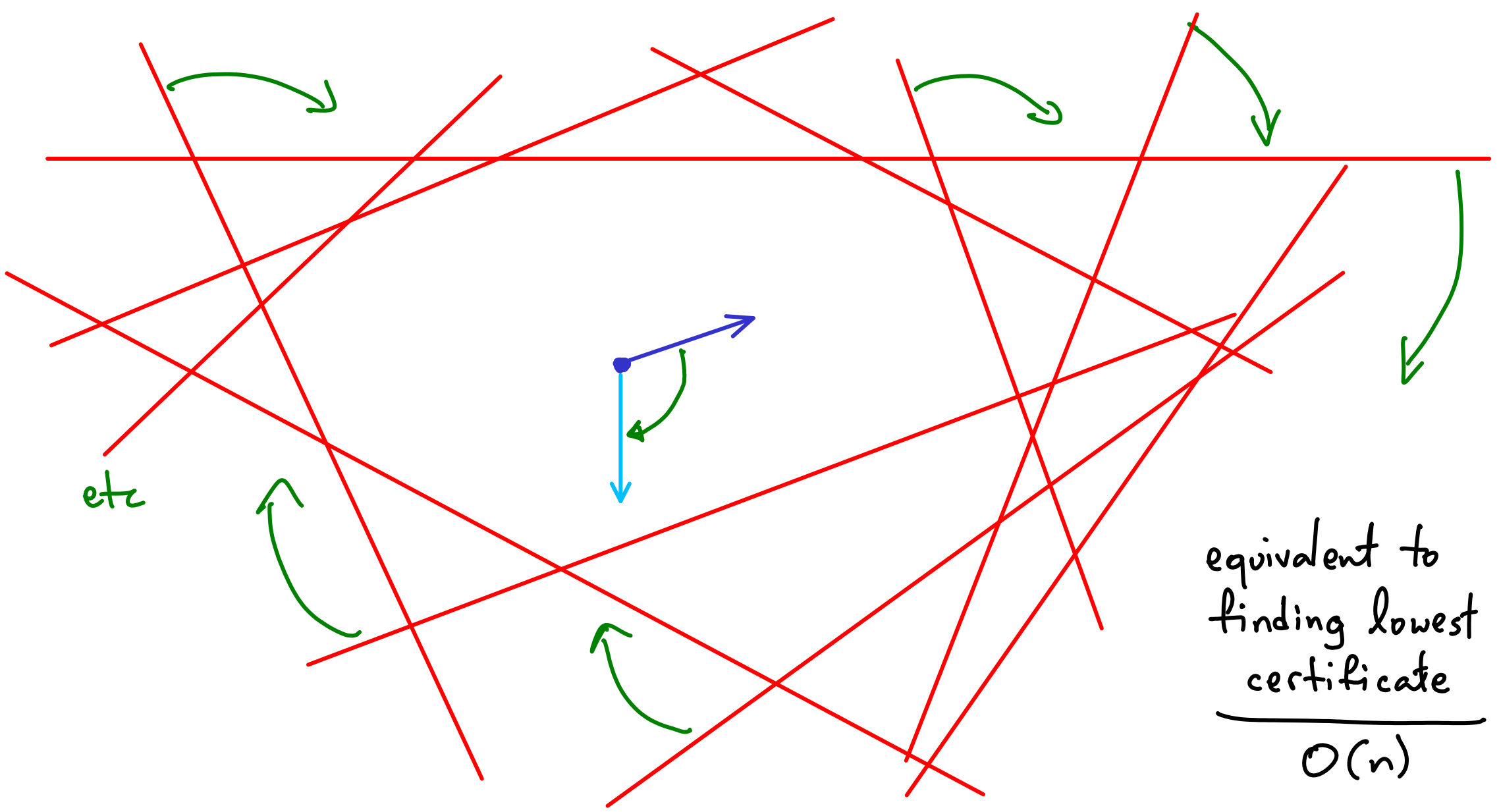
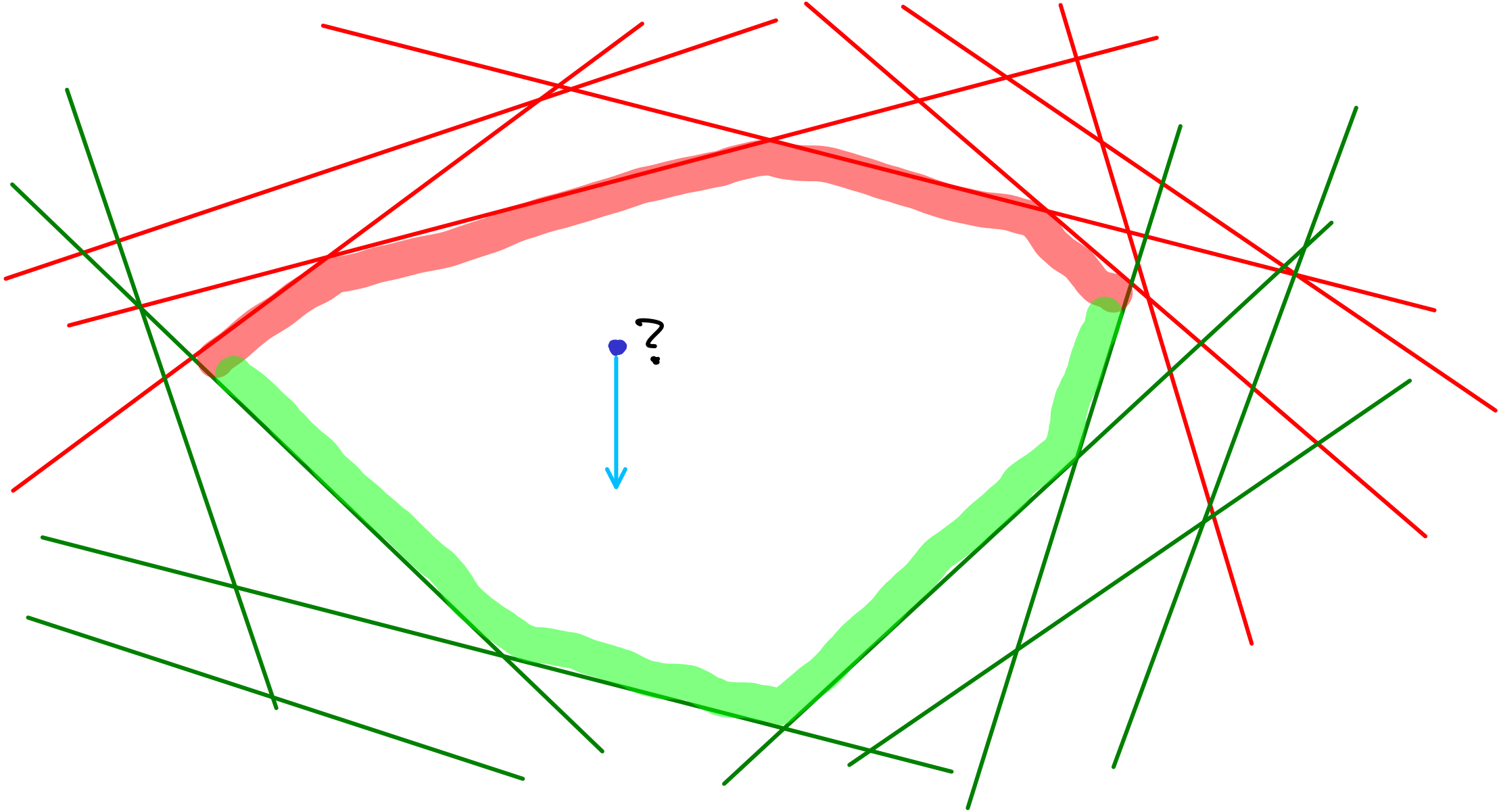
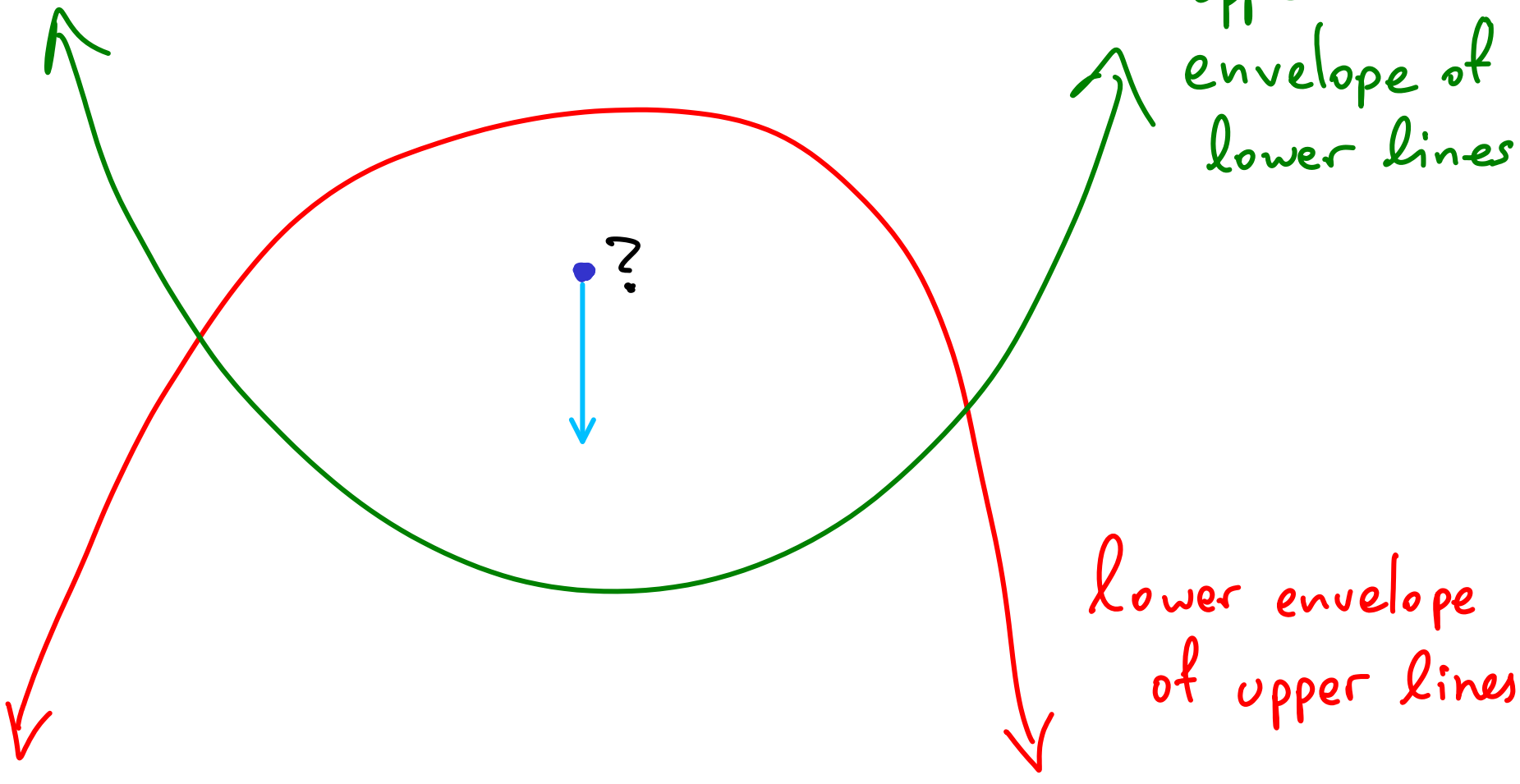
# FINDING * INTERSECTION OF HALF PLANES

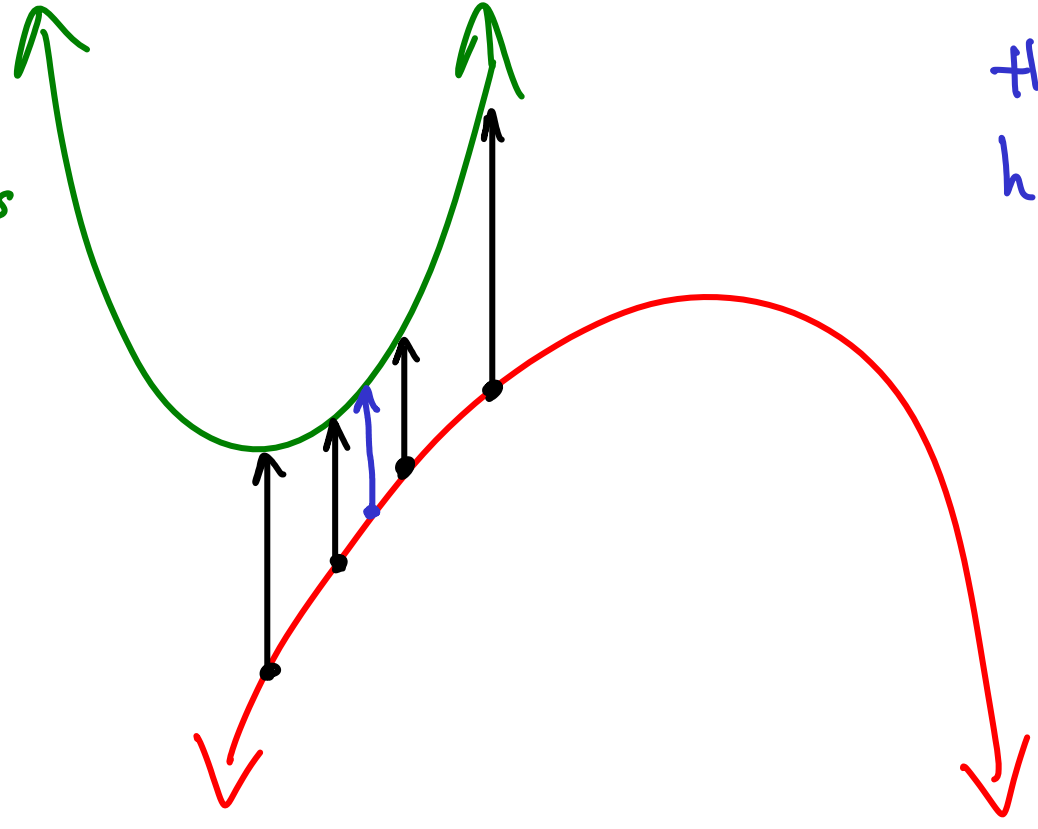
* = a point inside the intersection


etc
equivalent to finding lowest certificate
O(n)




upper
envelope of
lower lines
?
lower envelope
of upper lines

upper envelope of lower lines



lower envelope of upper lines

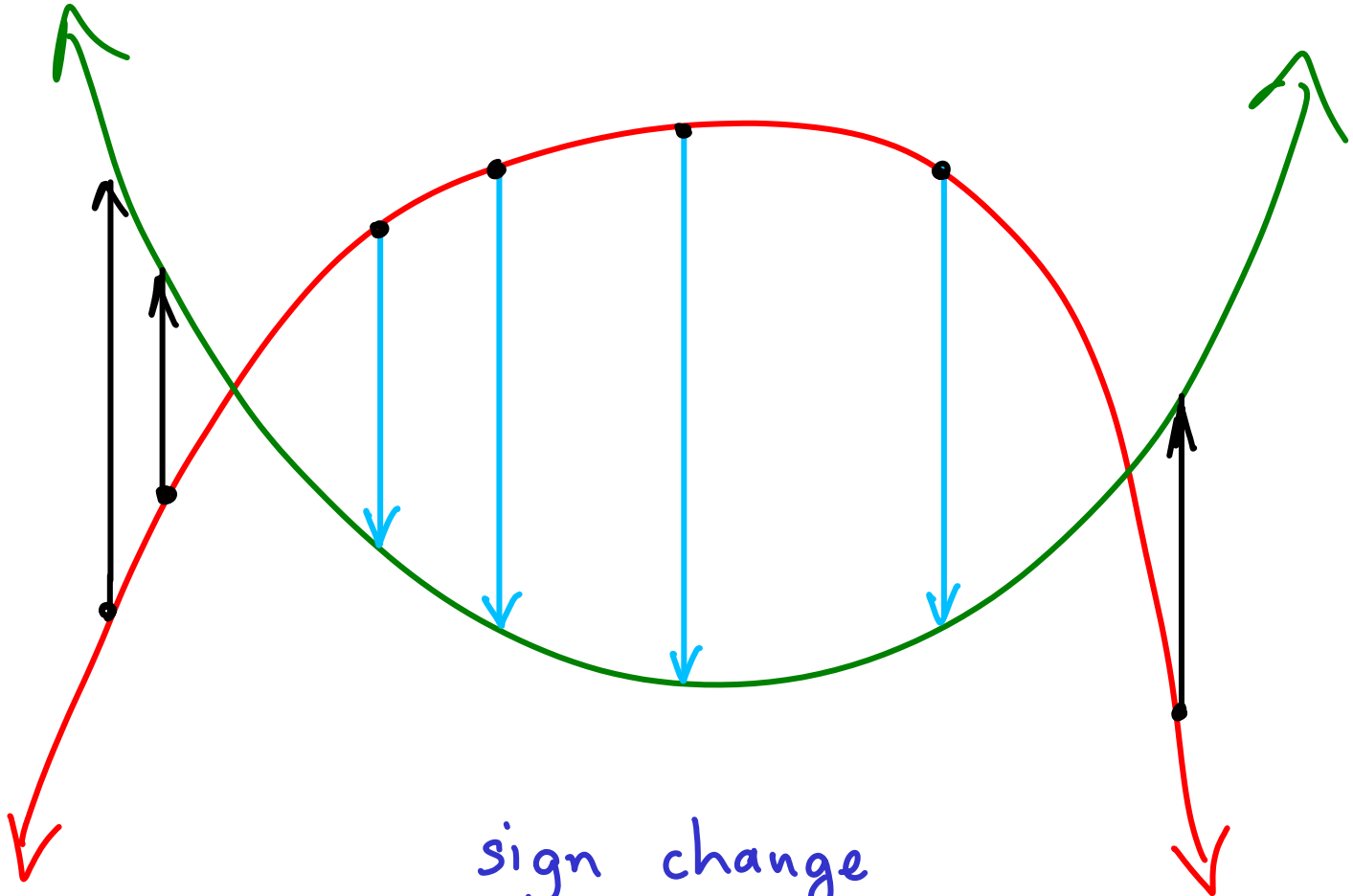
if no intersection then envelopes have some vertical separation that has a local min.

The algorithm that follows would find this local min, as a certificate that the intersection of the halfplanes is empty.

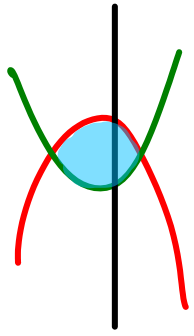
The local min is at a position where the green curve is parallel to the red curve

Of course, these are polygonal, so "parallel" could be a vertex vs an edge

measuring "red-to-gren" distance.



sign change
iff $\cap \neq \emptyset$



test line

O(n) time

This is the lowest red line (at this x-coordinate)

a

b

b'<0

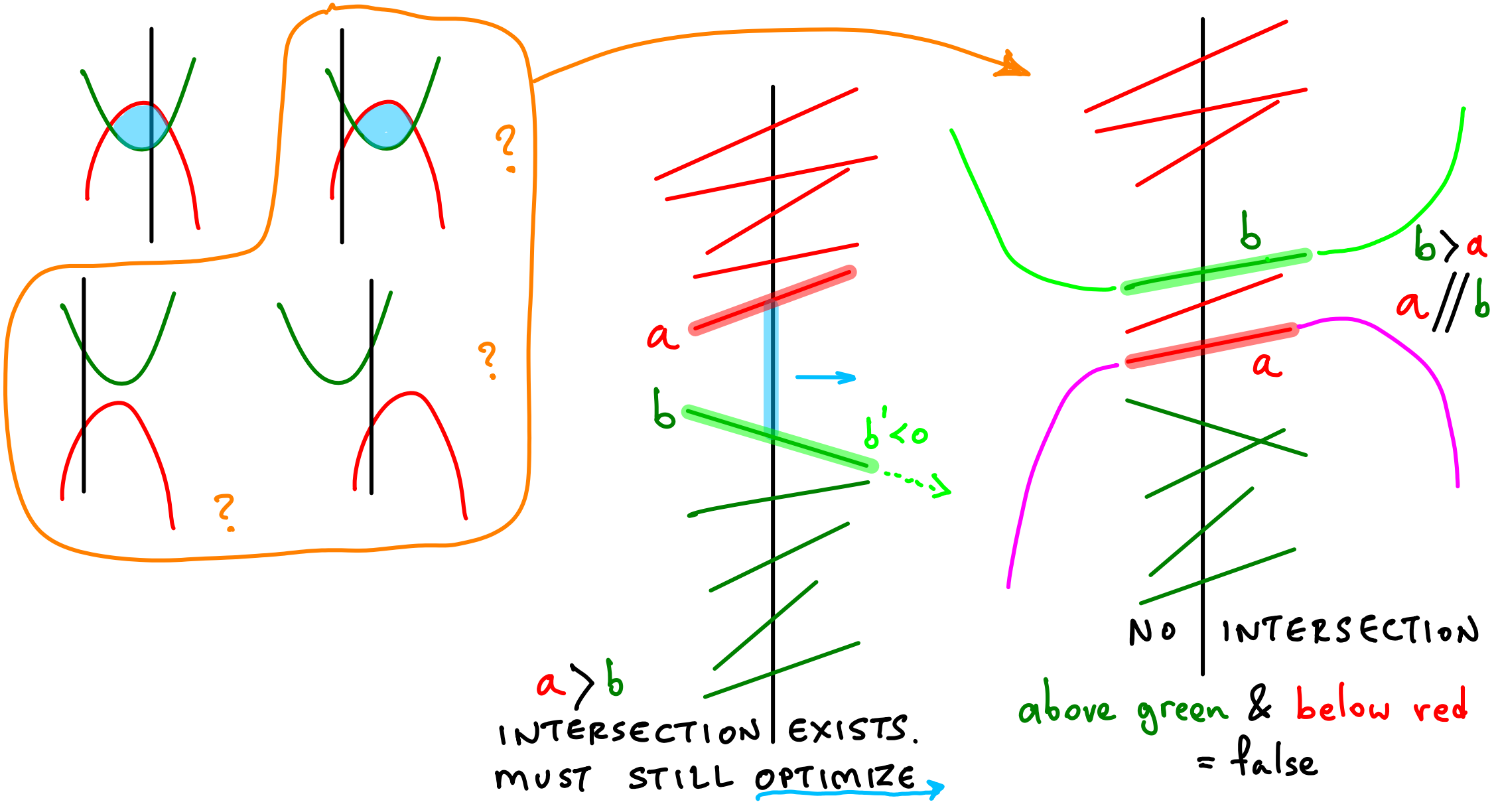
This is the highest green line

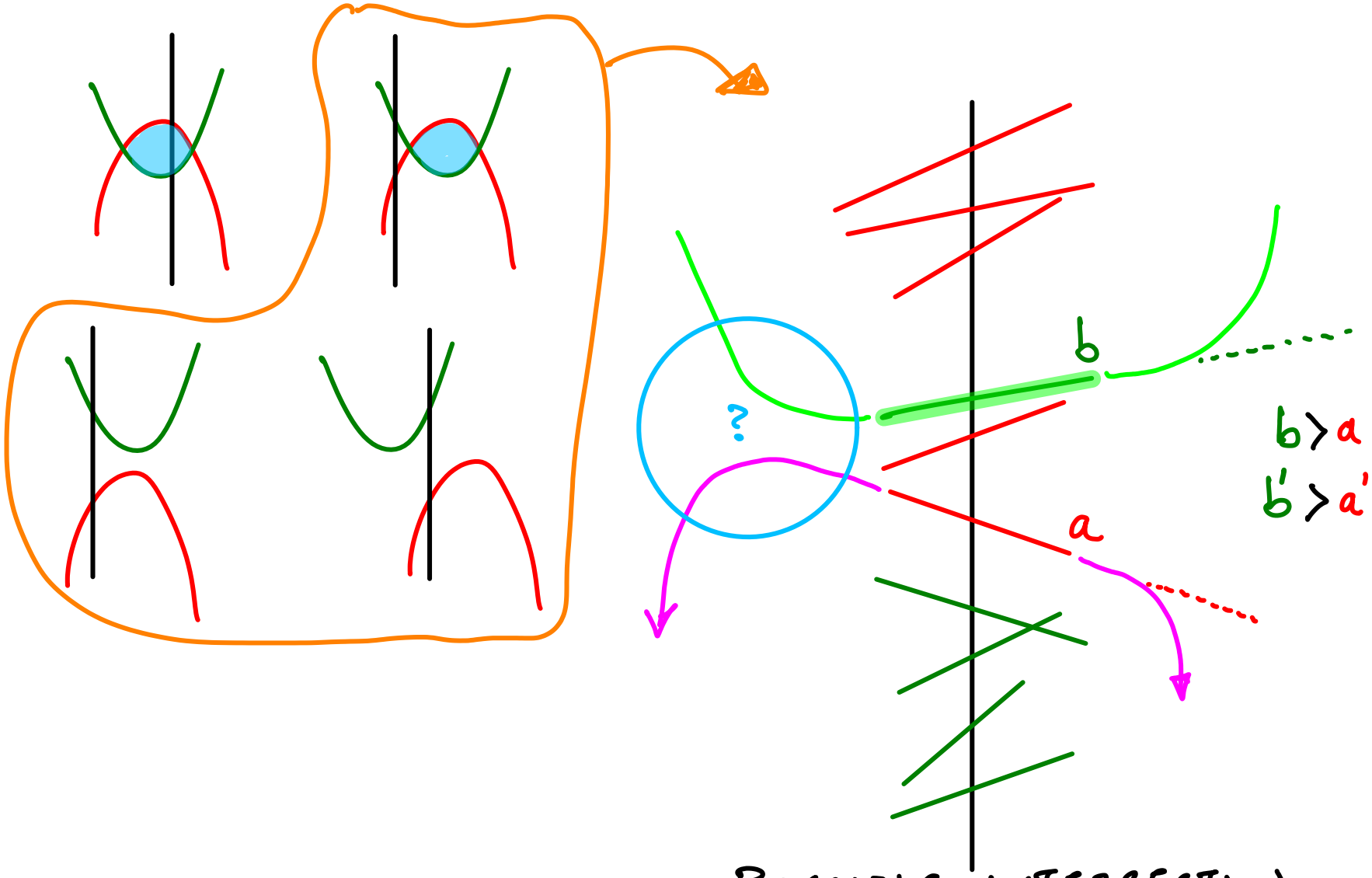
a > b

INTERSECTION EXISTS.

MUST STILL OPTIMIZE

DONE, if no optimization


?
?
?
a
b
b'<0
a>b
INTERSECTION EXISTS.
MUST STILL OPTIMIZE
b
b>a
a//b
a
NO INTERSECTION
above green & below red
= false


?
b
a
b > a
b' > a'
b' < a' symmetric
POSSIBLE INTERSECTION
TO THE LEFT

GIVEN A VERTICAL TEST LINE

WE CAN EITHER DECIDE THAT NO INTERSECTION EXISTS

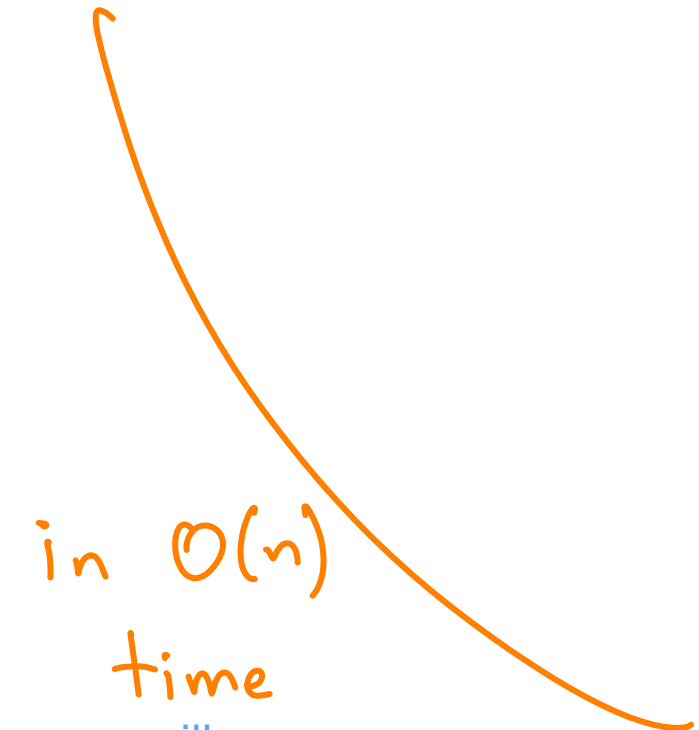
OR

FIND AN INTERSECTION

(AND IF NECESSARY ALSO
FIND WHICH SIDE MIN. IS ON)

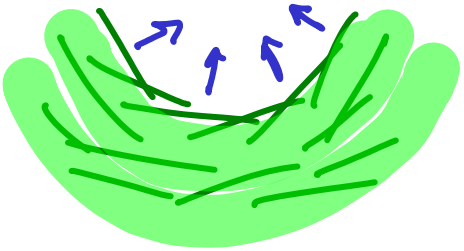
OR

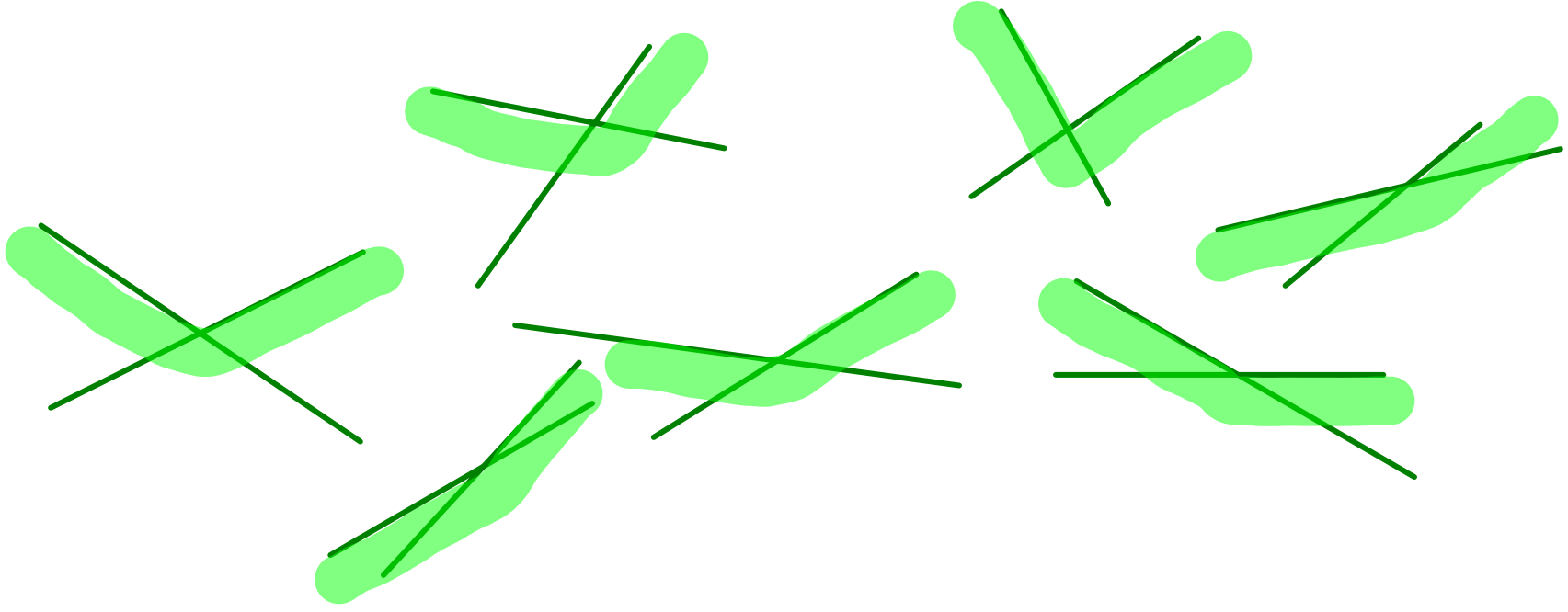
NOT KNOW ANSWER BUT AT LEAST
KNOW WHAT SIDE IT MIGHT BE ON




as we will see

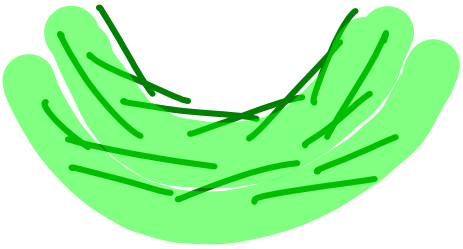
ARBITRARILY PAIR UP THE LOWER LINES

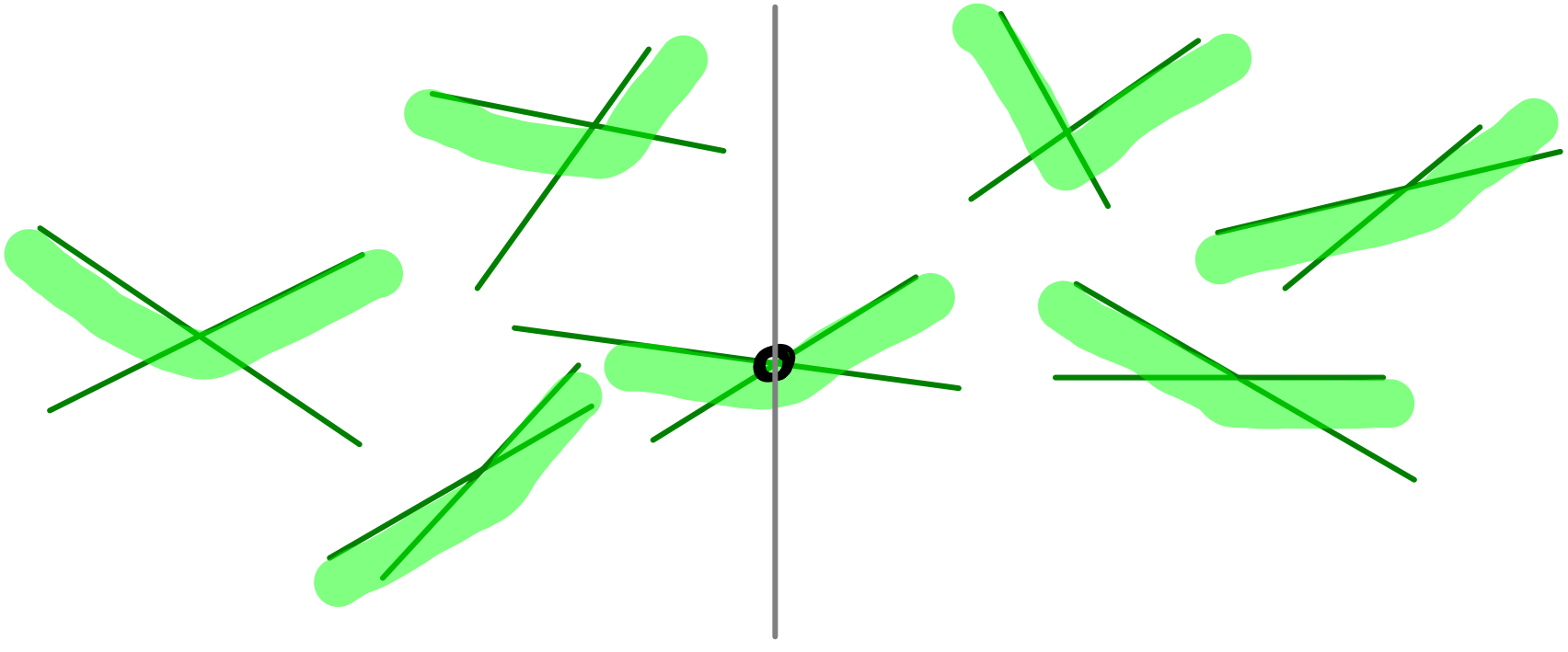




ARBITRARILY PAIR UP THE LOWER LINES

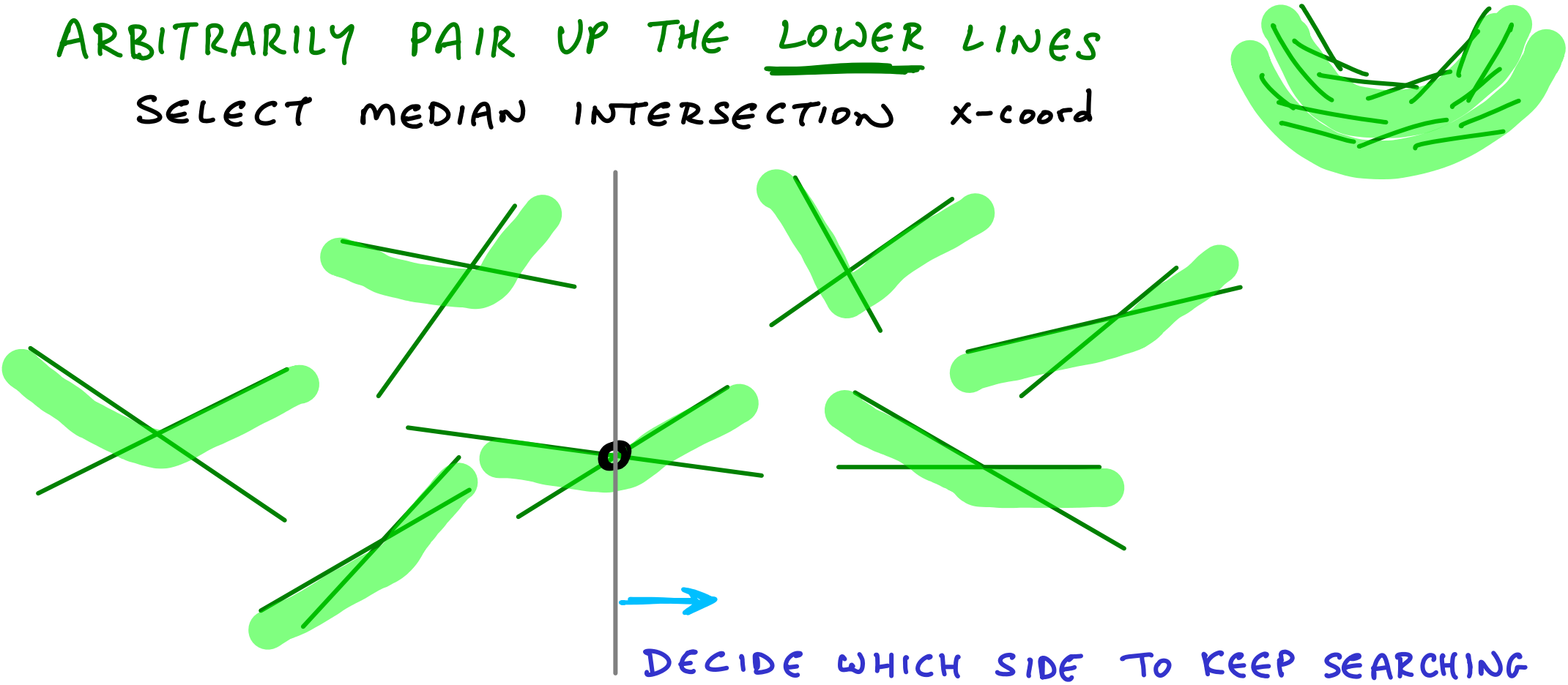
SELECT MEDIAN INTERSECTION x-coord





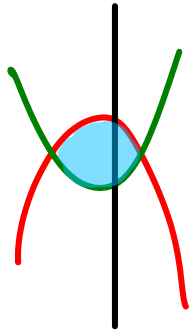
ARBITRARILY PAIR UP THE LOWER LINES

SELECT MEDIAN INTERSECTION x-coord

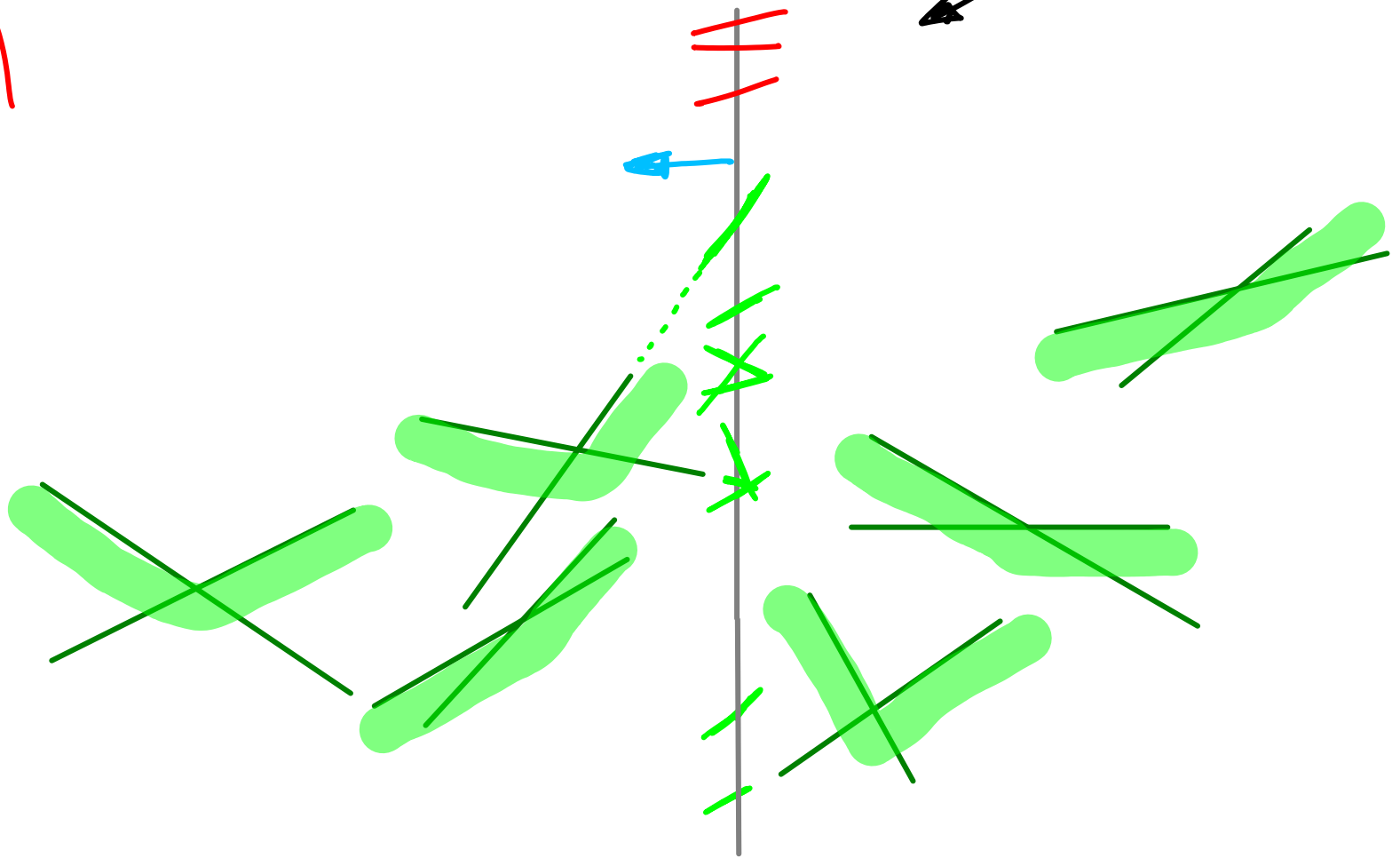


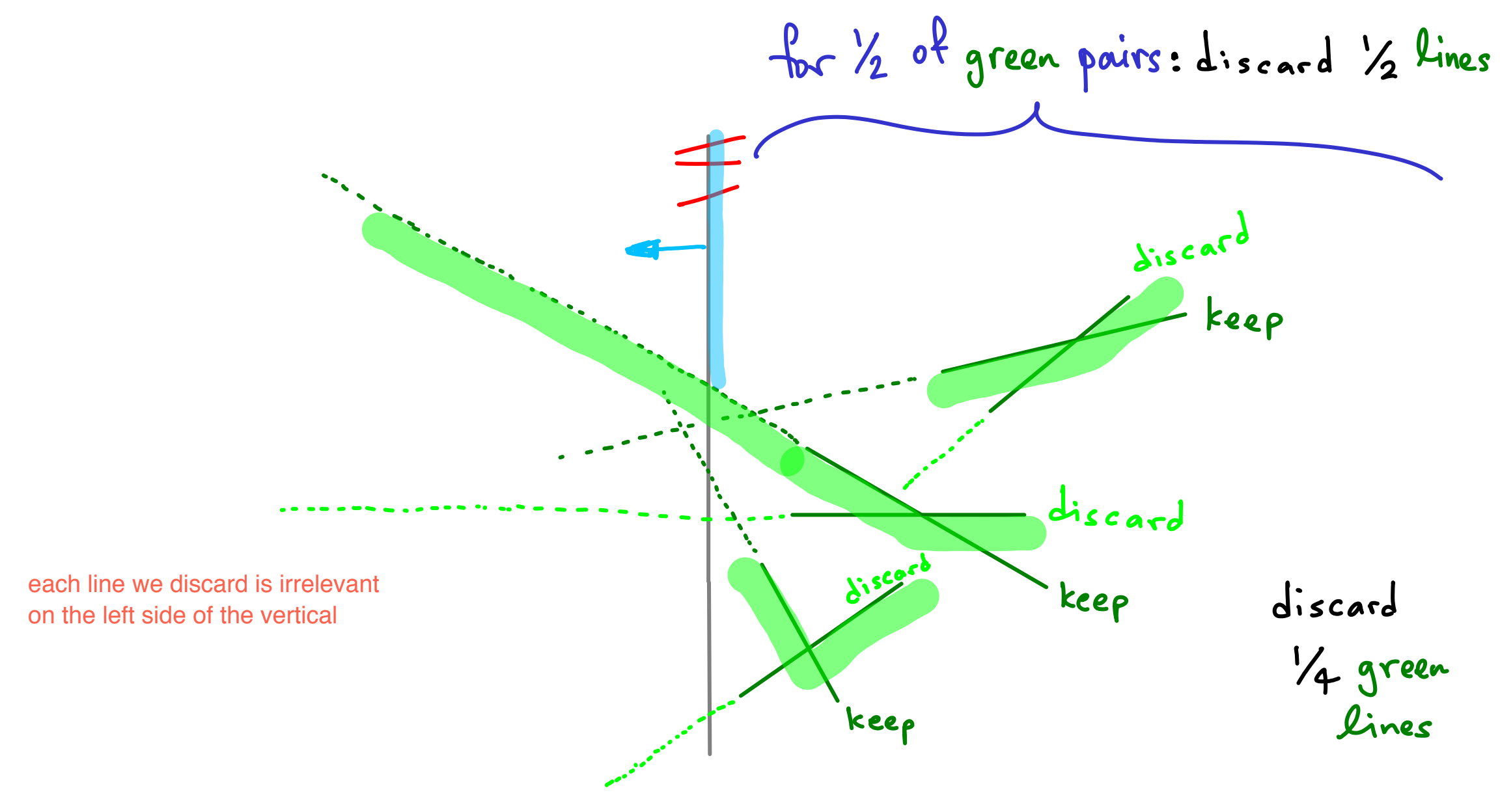
DECIDE WHICH SIDE TO KEEP SEARCHING

(if not done)



e.g. found intersection but want to optimize

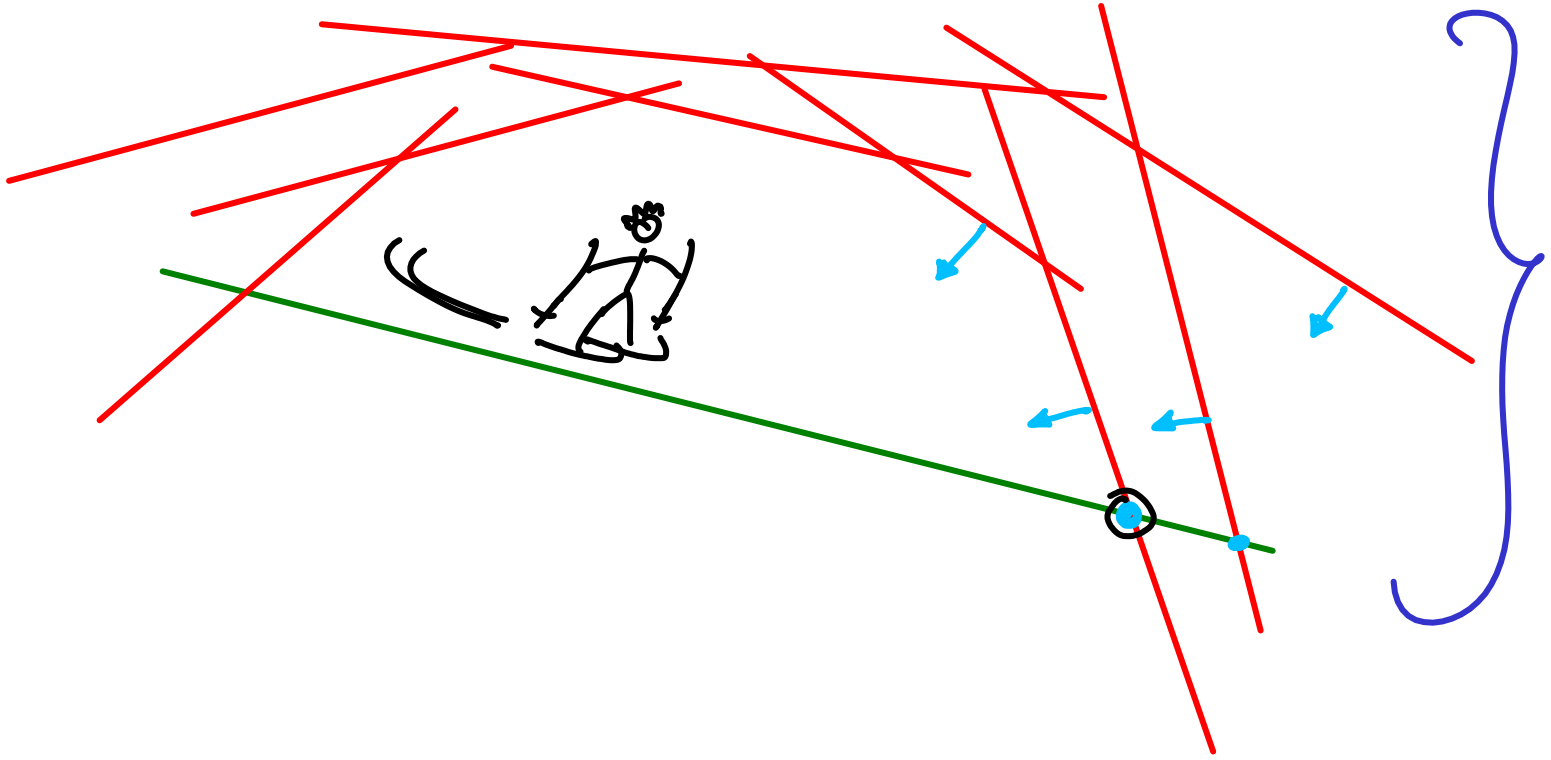



for ½ of green pairs: discard ½ lines
discard
keep
discard
discard
keep
keep
each line we discard is irrelevant
on the left side of the vertical
discard
¼ green
lines

Not good to just keep discarding green : get $O(\log n)$ rounds,
but keep doing $\Theta(n)$ work (still have lots of red)

Could take turns on each set, or pair up by color & get median of union.

FINALLY, ONE OF THE TWO GROUPS WILL BECOME 1 LINE



JUST COMPUTE ALL INTERSECTIONS OF STEEPER UPPER LINES & PICK HIGHEST

~1D PROBLEM

$O(n)$

# RECAP

0) ROTATE COORDINATE SYSTEM

1) PAIR LINES upper w/ upper ; lower w/ lower

2) COMPUTE MEDIAN OF $n/2$ INTERSECTION POINTS

3) PERFORM VERTICAL TEST & FIND SIDE CONTAINING MIN

4) FOR PAIRS ON WRONG SIDE, DISCARD 1 LINE

5) IF STILL >1 LINE IN EACH SET,
REPEAT FROM (1) on 3/4 of the lines

6) EASY SOLUTION WHEN ONLY 1 LINE IN A SET.

$O(n)$

# LEMONADE STAND (linear programming)

- YOU HAVE ONE HOUR TO SELL YOUR PRODUCT
- ASSUME EVERYTHING YOU MAKE WILL BE SOLD
- 2 PRODUCTS : REGULAR LEMONADE & SPECIAL LEMONADE
- EVERYTHING IS FRESH : YOU SQUEEZE LEMONS ON THE SPOT
  - ↳ IT TAKES TIME TO MAKE LEMONADE
- YOU HAVE A FIXED AMOUNT OF INGREDIENTS
- YOU WANT TO MAXIMIZE PROFIT DURING THIS HOUR
  - ↳ DON'T CARE ABOUT LEFTOVER INVENTORY

| | x | y |
|---|---|---|
| | REGULAR | SPECIAL |
| PROFIT | 1 | 2 |

maximize $x + 2y$

SUPPLIES → used

↓

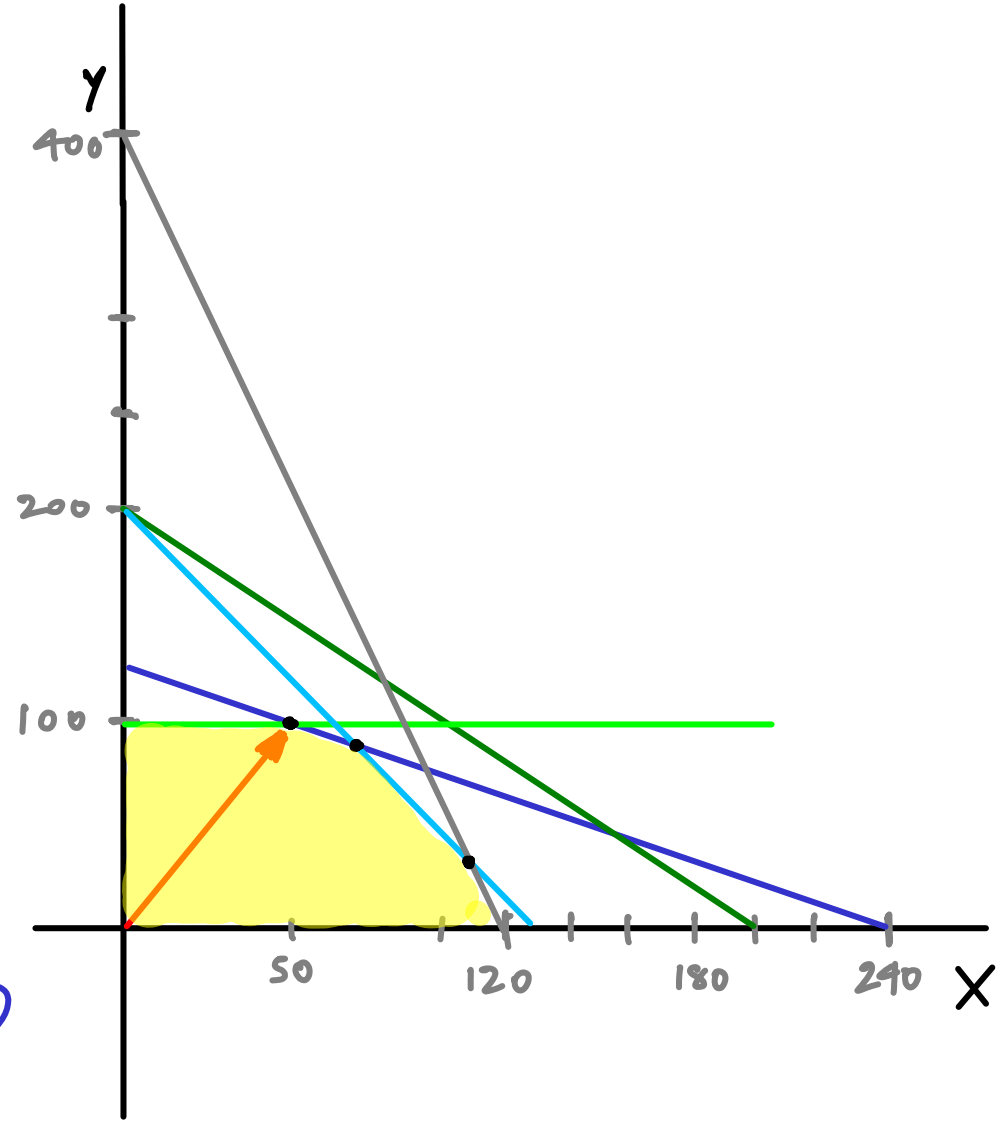
available

subject to :

| | | | | | |
|---|---|---|---|---|---|
| TIME | 60 | $\geq$ | $0.25x$ | $+$ | $0.5y$ |
| LEMONS | 200 | $\geq$ | $1x$ | $+$ | $1y$ |
| SUGAR | 250 | $\geq$ | $2x$ | $+$ | $1.25y$ |
| WATER | 240 | $\geq$ | $2x$ | $+$ | $0.6y$ |
| VODKA | 50 | $\geq$ | $0$ | | $0.5y$ |

$x, y \geq 0$

LINEAR PROGRAMMING



| | x REGULAR | y SPECIAL |
|---|---|---|
| PROFIT | 1 | 2 |

maximize $x + 2y$

SUPPLIES → used

↓

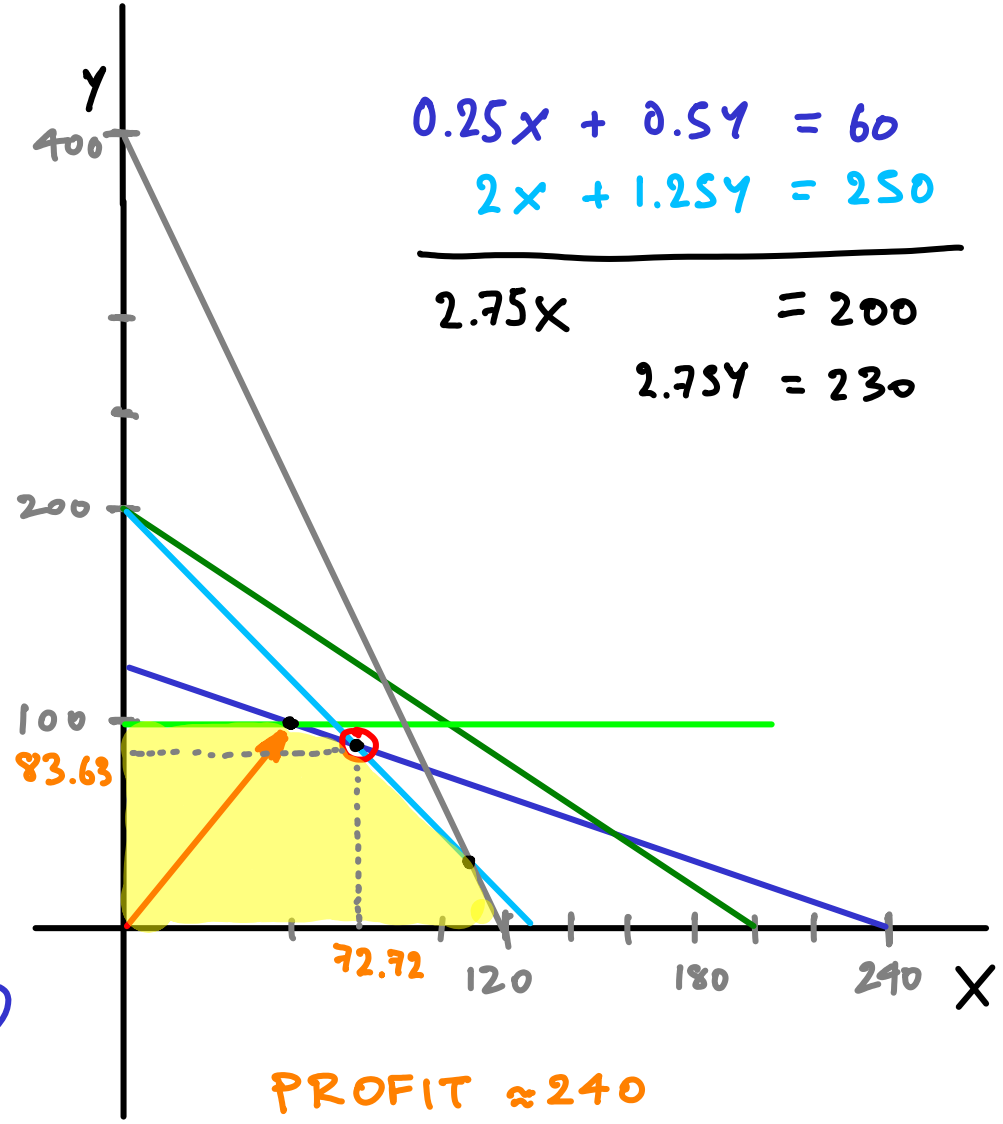
available

subject to :

| | | | | | |
|---|---|---|---|---|---|
| * TIME | 60 | ≥ | $0.25x$ | + | $0.5y$ |
| LEMONS | 200 | ≥ | $1x$ | + | $1y$ |
| * SUGAR | 250 | ≥ | $2x$ | + | $1.25y$ |
| WATER | 240 | ≥ | $2x$ | + | $0.6y$ |
| VODKA | 50 | ≥ | $0$ | | $0.5y$ |

$x, y \geq 0$

LINEAR PROGRAMMING




$0.25x + 0.5y = 60$

$2x + 1.25y = 250$

$2.75x = 200$

$2.75y = 230$

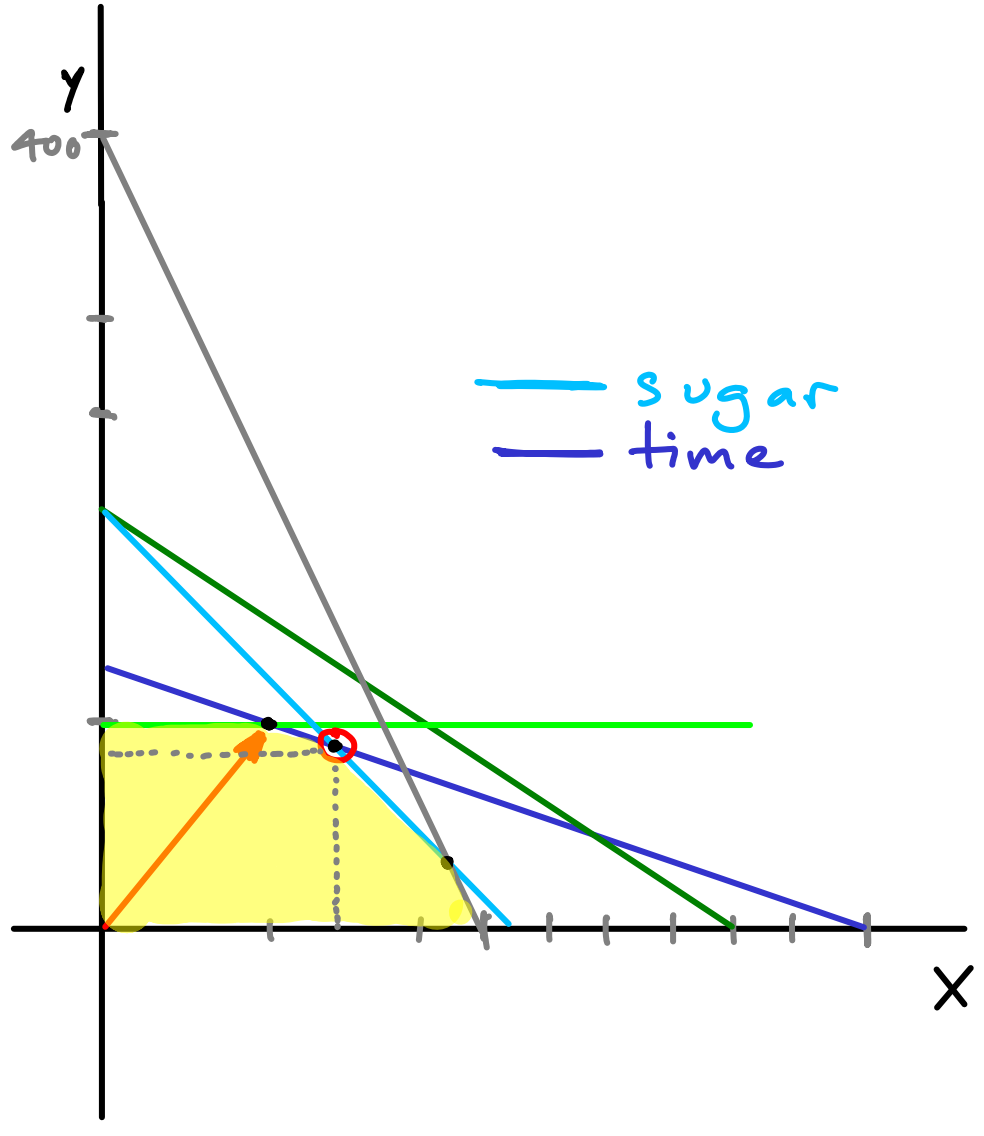
PROFIT ≈ 240

# Notes

Unless we are able to make
a special/original blend and
match the profit margin,
then we require an INTEGER solution

---

The solution tells us that we didn't
have enough time or sugar
↳ might be worth it to hire help
or make drinks less sweet



LP : maximize $aX + bY \left(+ cZ + \cdots + dX_d\right)$

subject to $r_1X + s_1Y \leq k_1$

$r_2X + s_2Y \leq k_2$

$\vdots$

$r_nX + s_nY \leq k_n$

$d$ = dimension

normally $n \geq d$