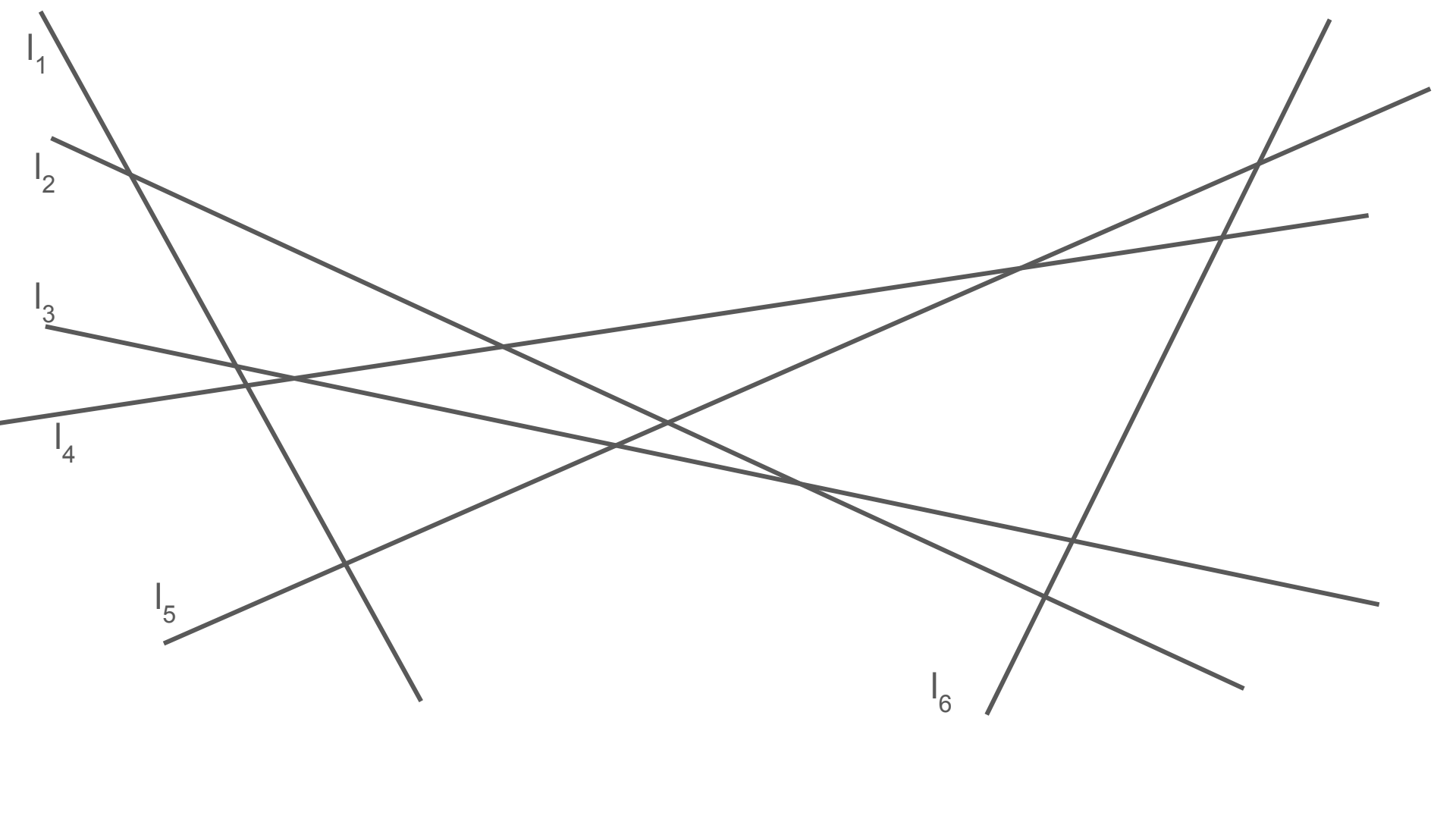
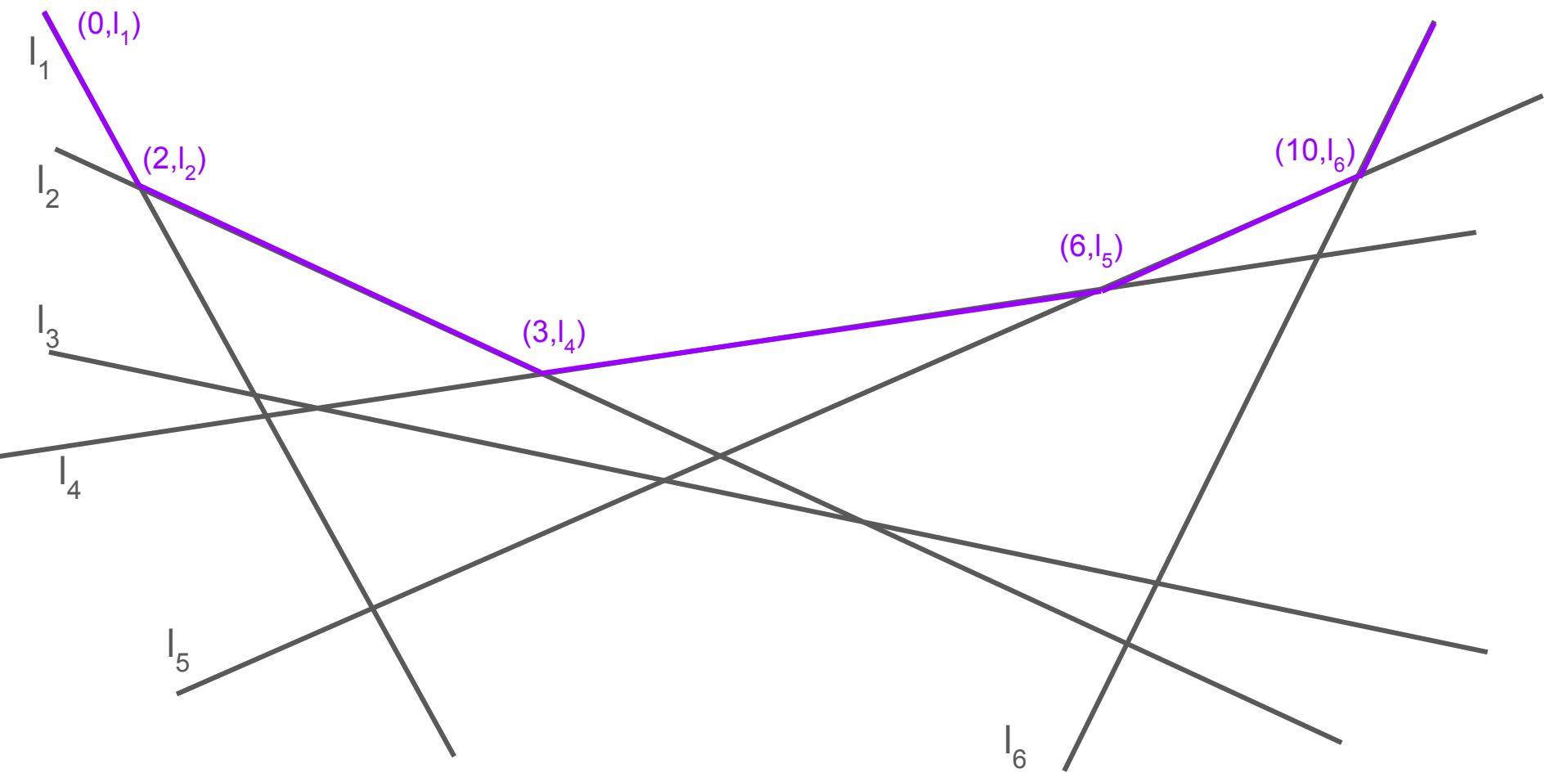
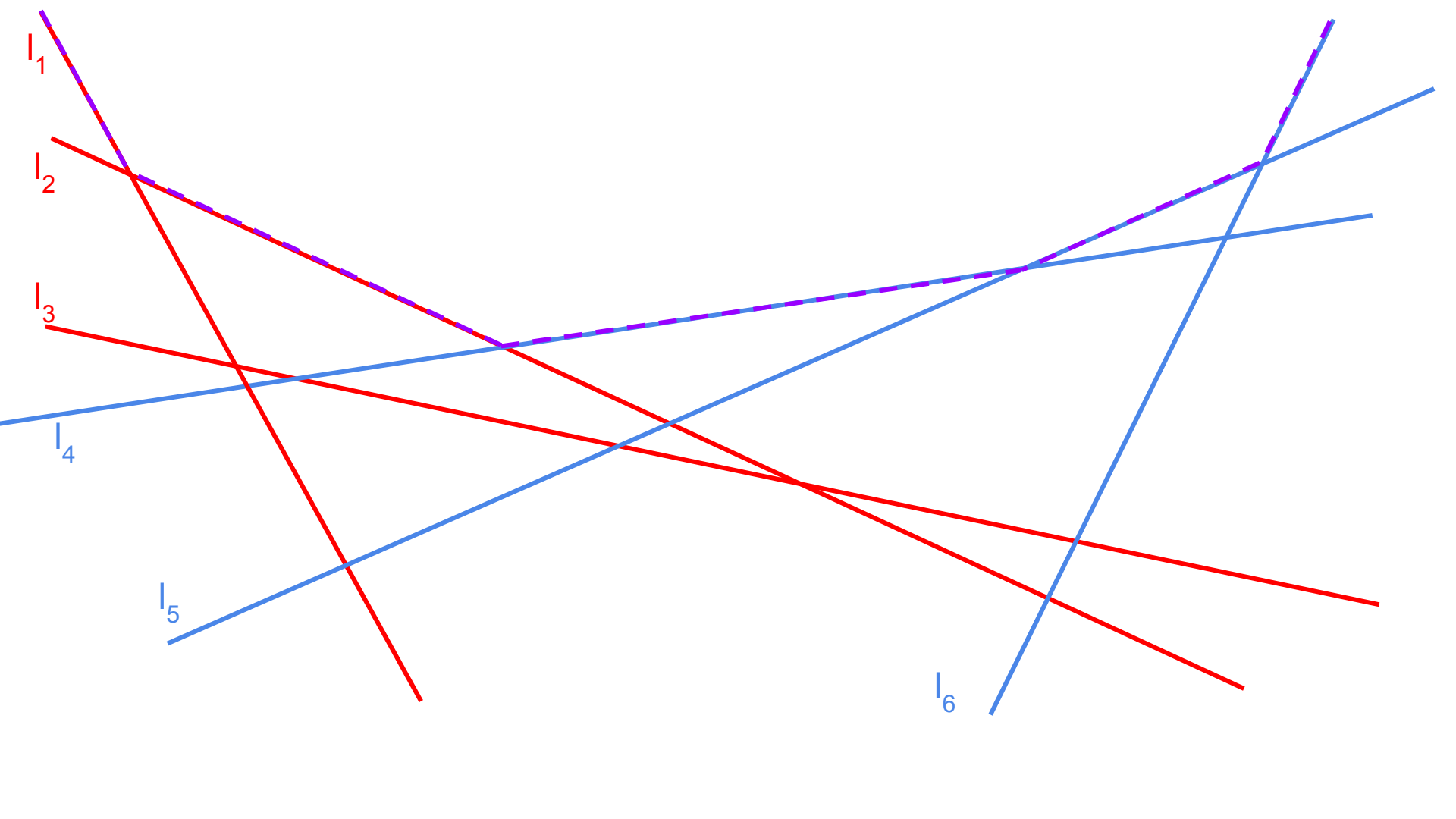
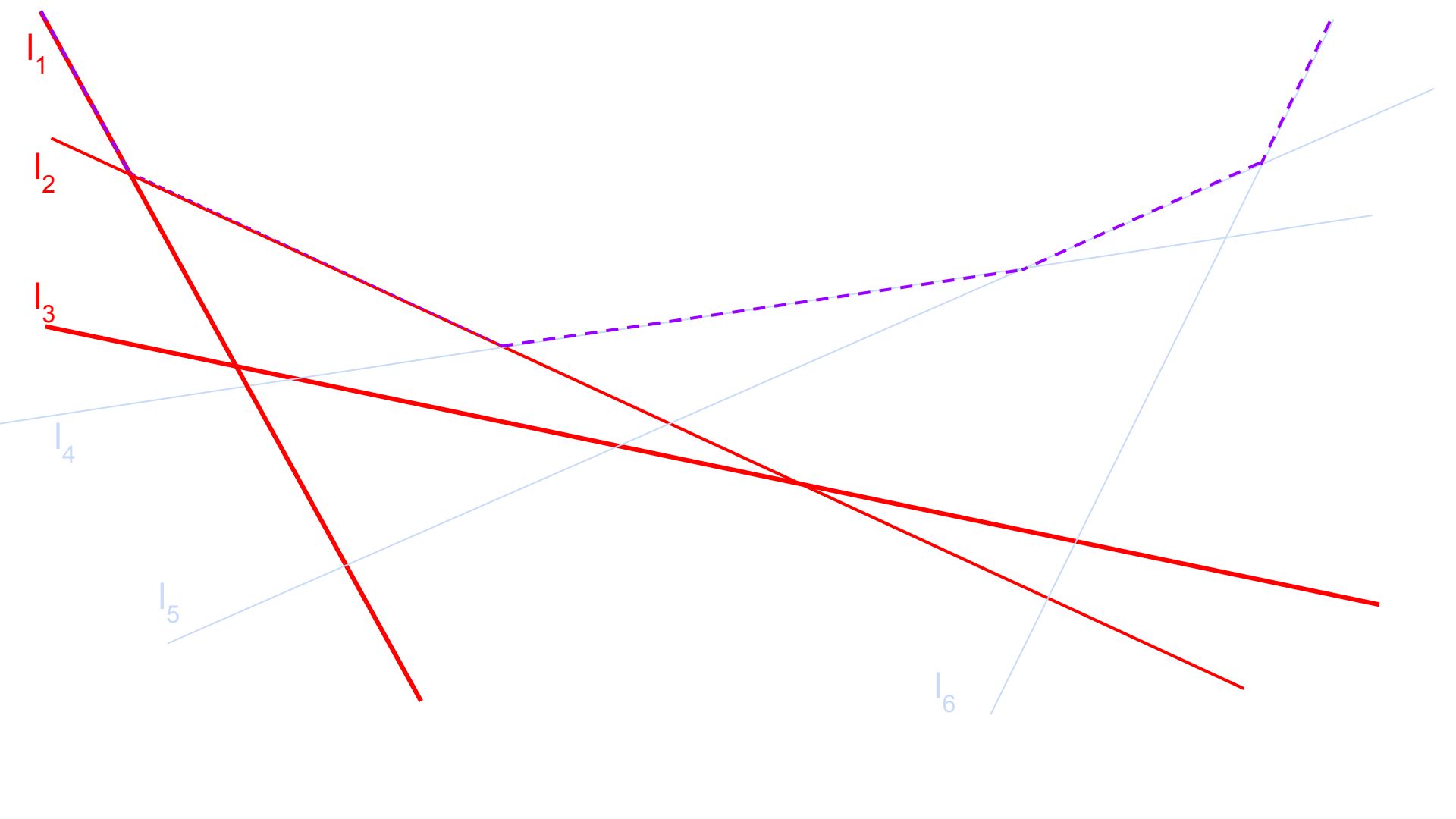


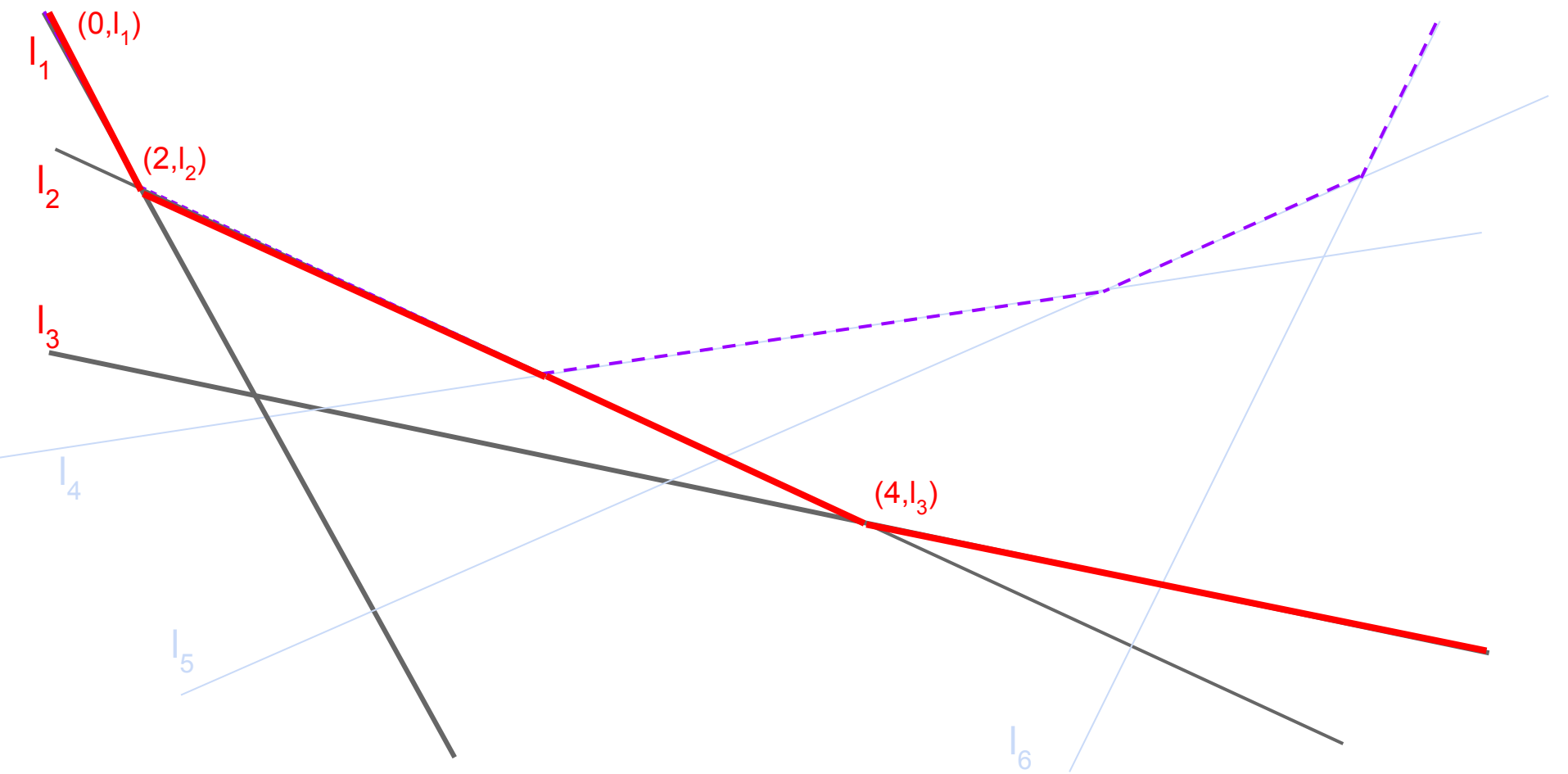


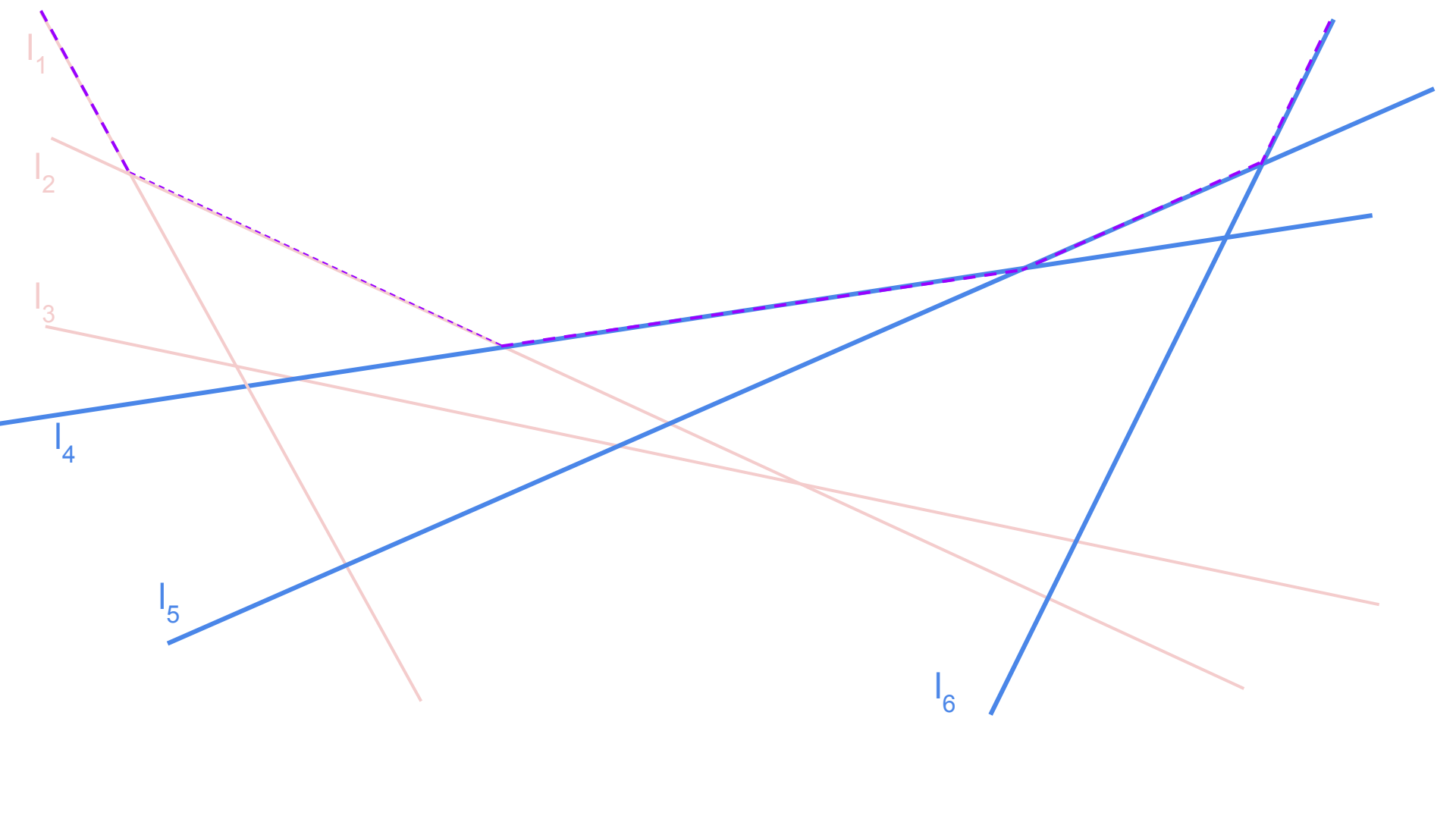


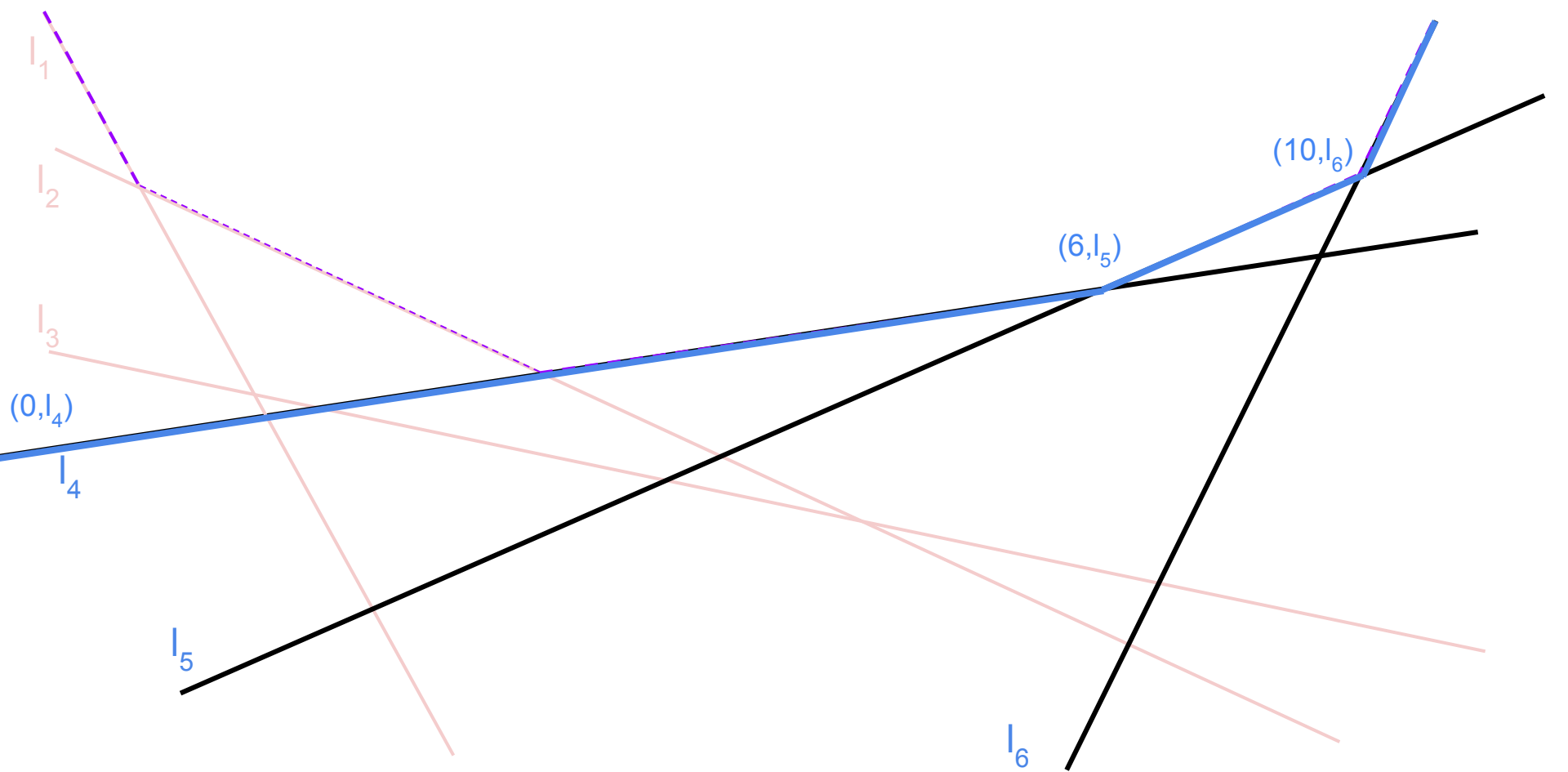


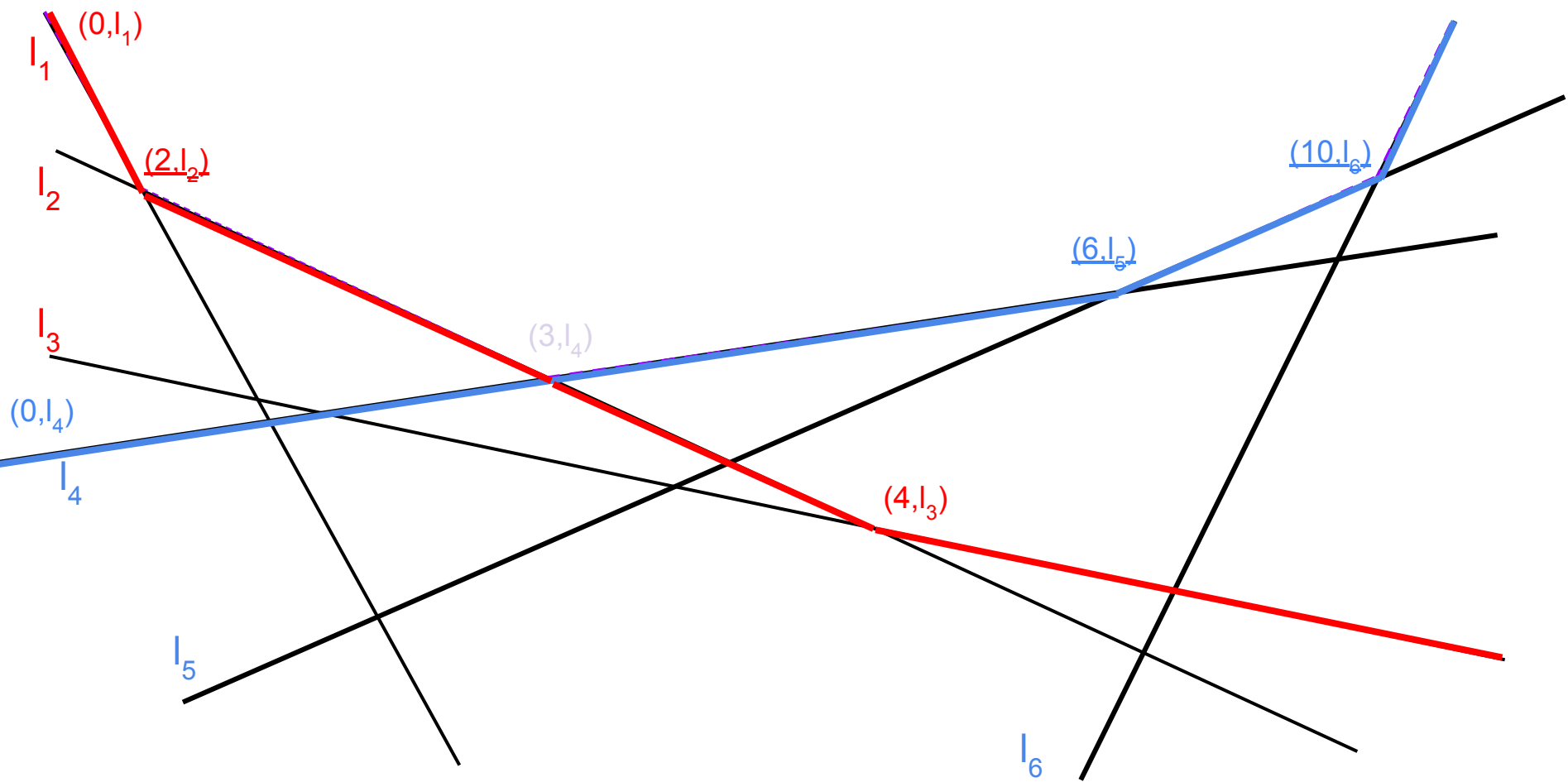




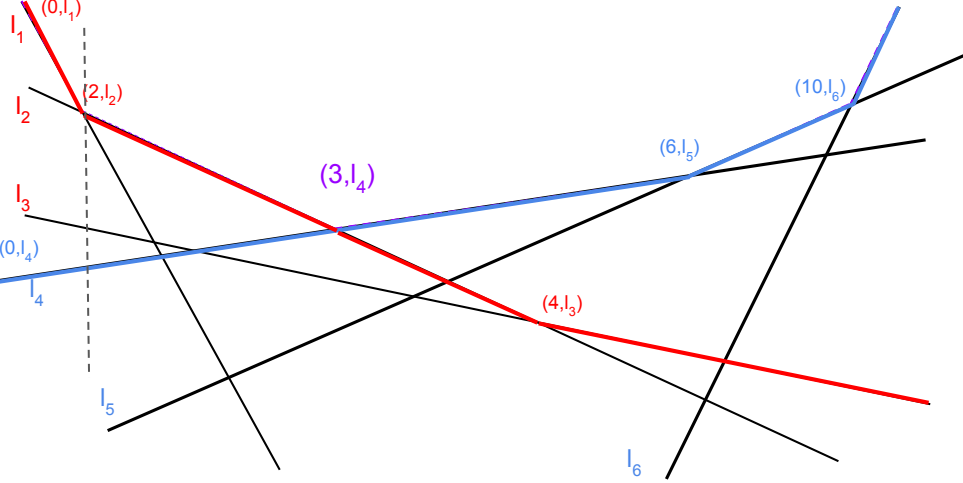






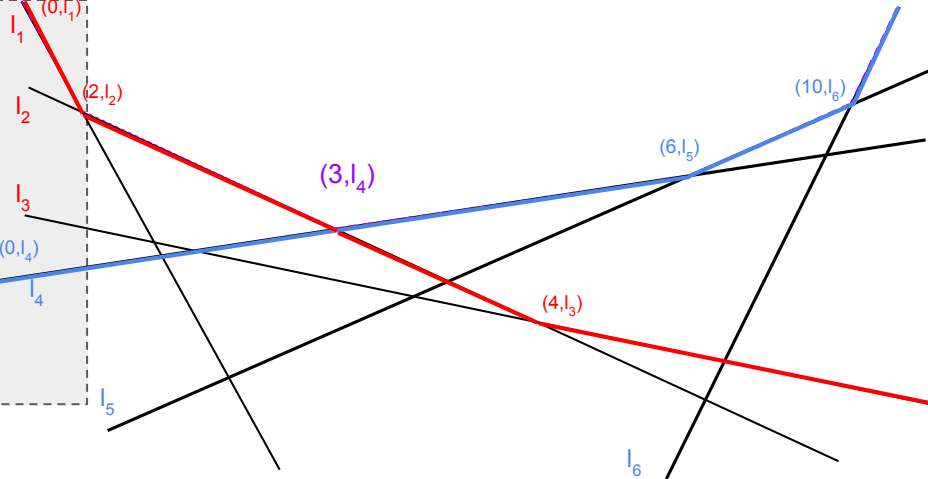


Insight: the true upper envelope $U(x) = \max(U_1(x), U_2(x))$



How do I know when lines cross?

Say I know the curr lines are $I_2 = -\frac{1}{2}x + 2$ and $I_4 = x - 1$



$[prevT, t] = [0, 2]$

$currRed = l_1$

$currBlue = l_4$

Top = red

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed$ = current line from red envelope

$currBlue$ = current line from blue recursive envelope

examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:

Add whichever is higher

if $currRed$ and $currBlue$ cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$prevT = t$

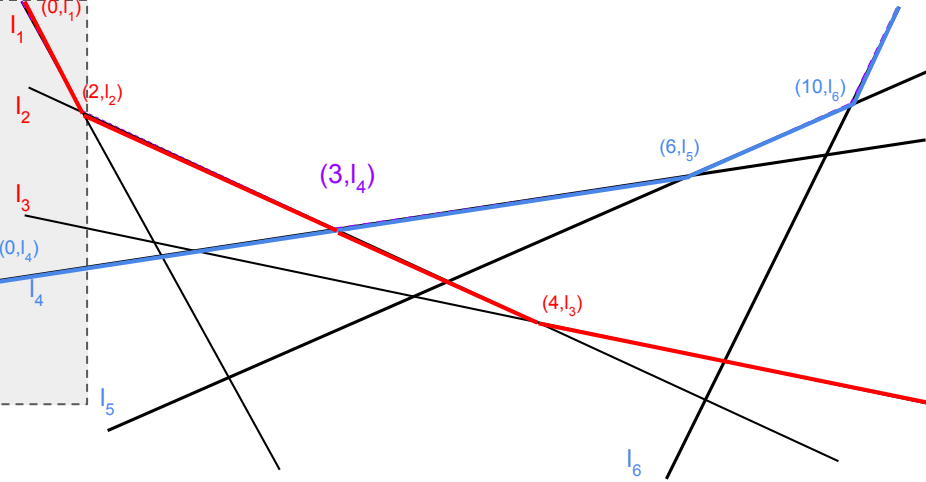
$(2, l_2)$

$(4, l_3)$

$(6, l_5)$

$(10, l_6)$

$(3, l_4)$



$[prevT, t] = [0, 2]$
 $currRed = l_1$
 $currBlue = l_4$
 Top = red

$(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed$ = current line from red envelope

$currBlue$ = current line from blue recursive envelope

examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:

Add whichever is higher

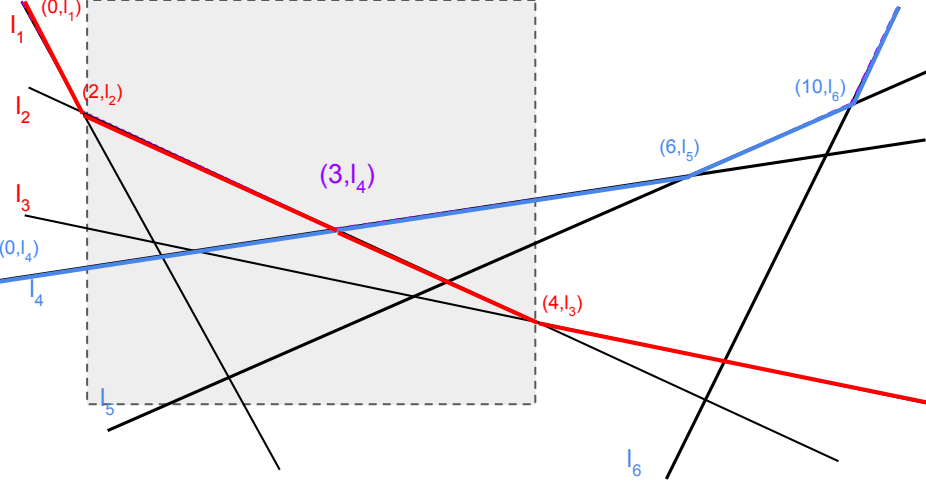
if $currRed$ and $currBlue$ cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$prevT = t$

Res: $(0, l_1)$



$[\text{prevT}, t] = [2, 4]$
 $\text{currRed} = l_2$
 $\text{currBlue} = l_4$
 Top = red

$(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$\text{currRed} = \text{current line from red envelope}$

$\text{currBlue} = \text{current line from blue recursive envelope}$

examine currRed and currBlue on the interval $[\text{prevT}, t]$

if currRed and currBlue do not cross:

Add whichever is higher

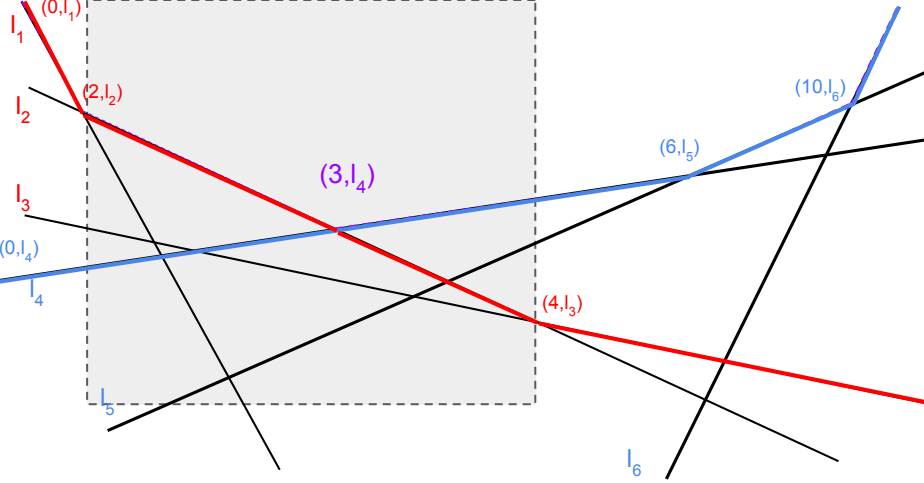
if currRed and currBlue cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$\text{prevT} = t$

Res: $(0, l_1)$



(they dont cross at $t = 2$ YET)

$[prevT, t] = [2, 4]$

$currRed = l_2$

$currBlue = l_4$

Top = red

$(2, l_2)$

$(4, l_3)$

$(6, l_5)$

$(10, l_6)$

$(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed$ = current line from red envelope

$currBlue$ = current line from blue recursive envelope

examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:

Add whichever is higher

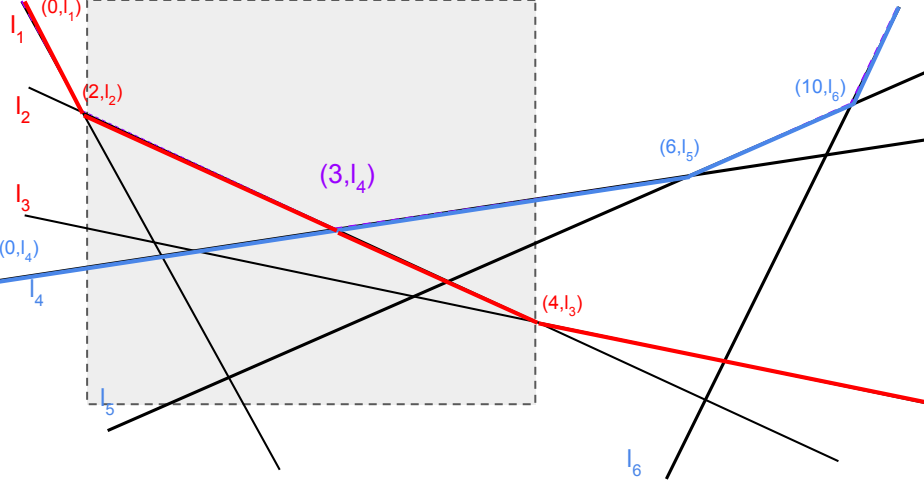
if $currRed$ and $currBlue$ cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$prevT = t$

Res: $(0, l_1)$ $(2, l_2)$



(they cross at $x = 3$)

$[\text{prevT}, t] = [2, 4]$

$\text{currRed} = l_2$

$\text{currBlue} = l_4$

Top = red

$(2, l_2)$

$(4, l_3)$

$(6, l_5)$

$(10, l_6)$

$(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$\text{currRed} = \text{current line from red envelope}$

$\text{currBlue} = \text{current line from blue recursive envelope}$

examine currRed and currBlue on the interval $[\text{prevT}, t]$

if currRed and currBlue do not cross:

Add whichever is higher

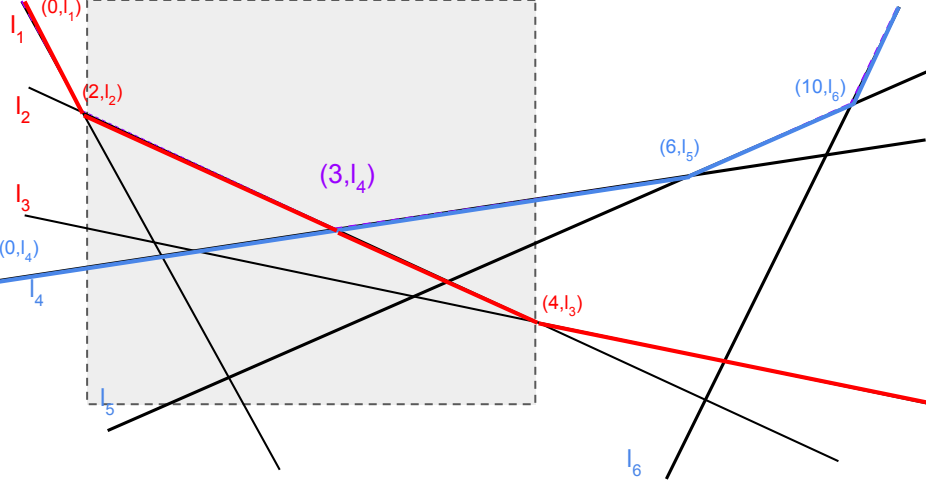
if currRed and currBlue cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$\text{prevT} = t$

Res: $(0, l_1)$ $(2, l_2)$



(they cross at $x = 3$)

$[\text{prevT}, t] = [2, 4]$

$\text{currRed} = l_2$

$\text{currBlue} = l_4$

Top = **blue**

$(2, l_2)$

$(4, l_3)$

$(6, l_5)$

$(10, l_6)$

$(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

currRed = current line from **red** envelope

currBlue = current line from **blue** recursive envelope

examine currRed and currBlue on the interval $[\text{prevT}, t]$

if currRed and currBlue do not cross:

Add whichever is higher

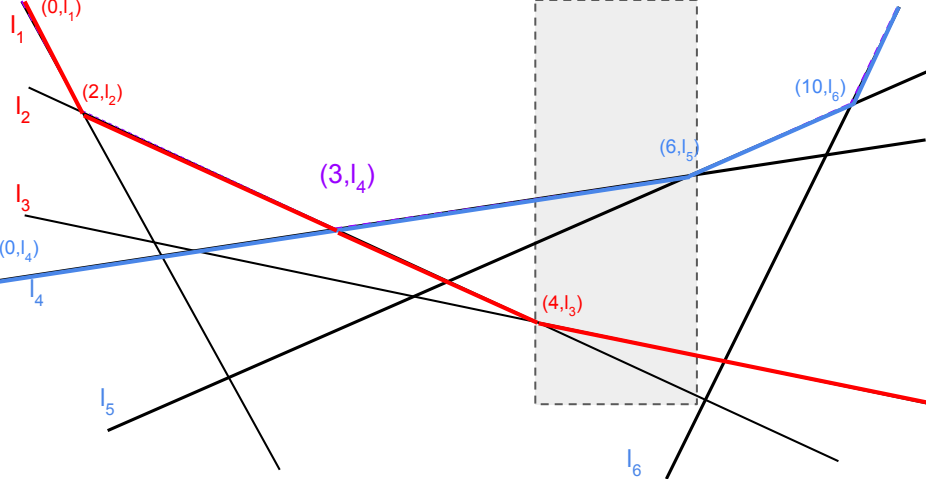
if currRed and currBlue cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$\text{prevT} = t$

Res: $(0, l_1)$ $(2, l_2)$ $(3, l_4)$



$[prevT, t] = [4, 6]$
 $currRed = l_3$
 $currBlue = l_5$
 Top = blue

$(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed = \text{current line from red envelope}$

$currBlue = \text{current line from blue recursive envelope}$

examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:

Add whichever is higher

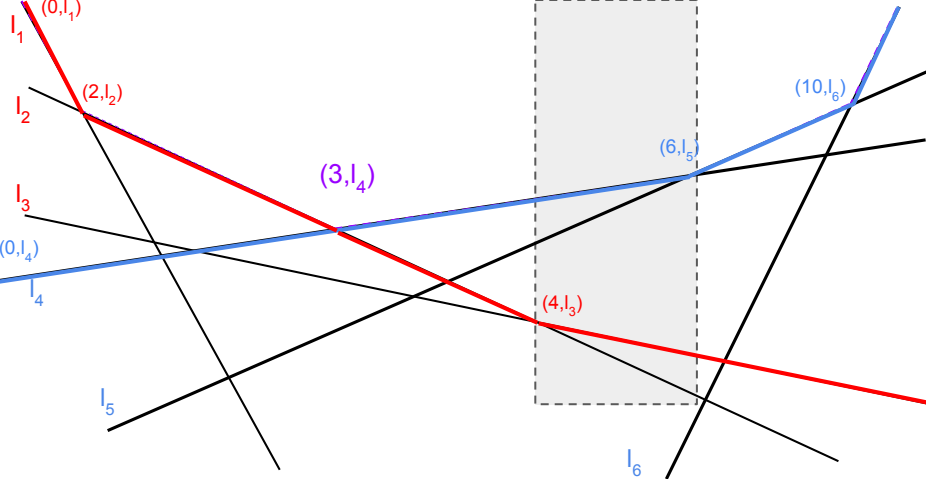
if $currRed$ and $currBlue$ cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$prevT = t$

Res: $(0, l_1)$ $(2, l_2)$ $(3, l_4)$ $(6, l_5)$



$[prevT, t] = [4, 6]$
 $currRed = l_3$
 $currBlue = l_5$
 Top = blue

$(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed = \text{current line from red envelope}$

$currBlue = \text{current line from blue recursive envelope}$

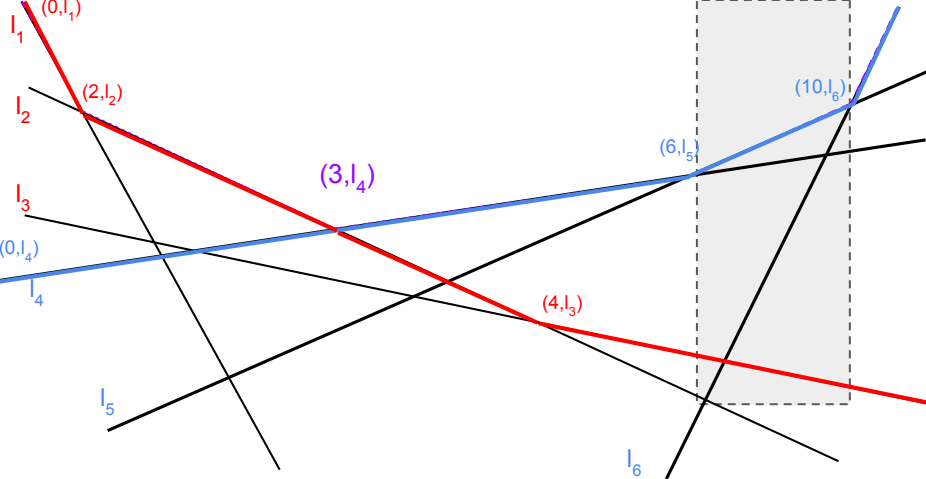
examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:
Add whichever is higher

if $currRed$ and $currBlue$ cross at x^* :
 add x^* as a breakpoint
 switch which recursive envelope is on top

$prevT = t$

Res: $(0, l_1)$ $(2, l_2)$ $(3, l_4)$ $(6, l_5)$



$[prevT, t] = [6, 10]$

$currRed = l_3$

$currBlue = l_6$

Top = blue

$(2, l_2)$

$(4, l_3)$

$(6, l_5)$

$(10, l_6)$

$(3, l_4)$

Scan left to right:

$t = \text{next red/blue breakpoint}$

$currRed$ = current line from red envelope

$currBlue$ = current line from blue recursive envelope

examine $currRed$ and $currBlue$ on the interval $[prevT, t]$

if $currRed$ and $currBlue$ do not cross:

Add whichever is higher

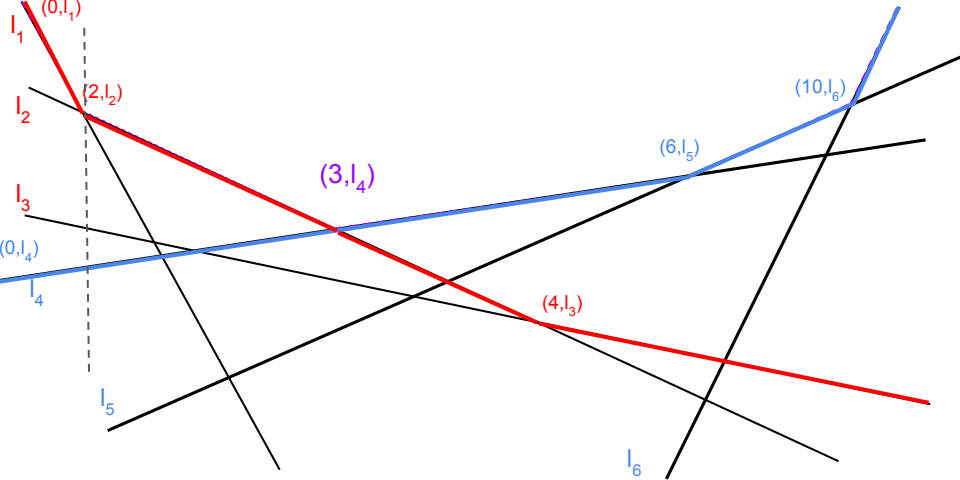
if $currRed$ and $currBlue$ cross at x^* :

add x^* as a breakpoint

switch which recursive envelope is on top

$prevT = t$

Res: $(0, l_1)$ $(2, l_2)$ $(3, l_4)$ $(6, l_5)$ $(10, l_6)$

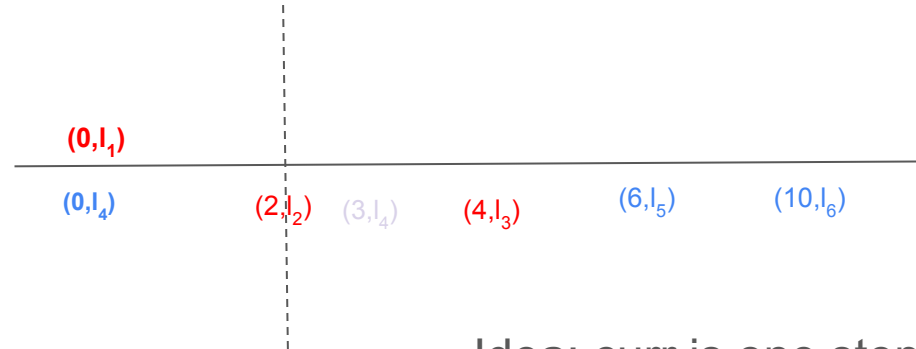


First Idea: Sweep a line left to right,
choose higher line based on curr breakpt

for curr = 2,...,10

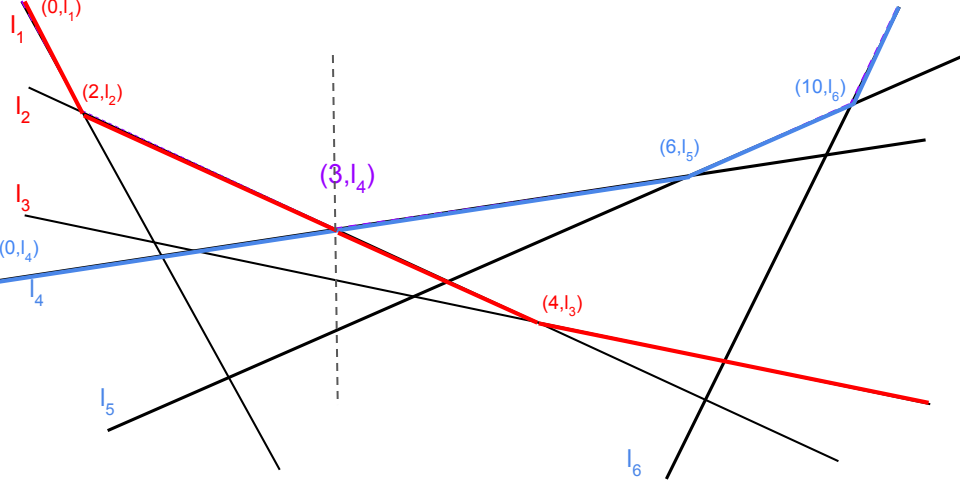
Curr = 2
currRed: I_2
currBlue: I_4

$(0, I_1)$ added since $I_1(0) > I_4(0)$



And we encounter a
new red break pt

Idea: curr is one step ahead of whatever breakpt we want to add



First Idea: Sweep a line left to right,
choose higher line based on curr breakpt

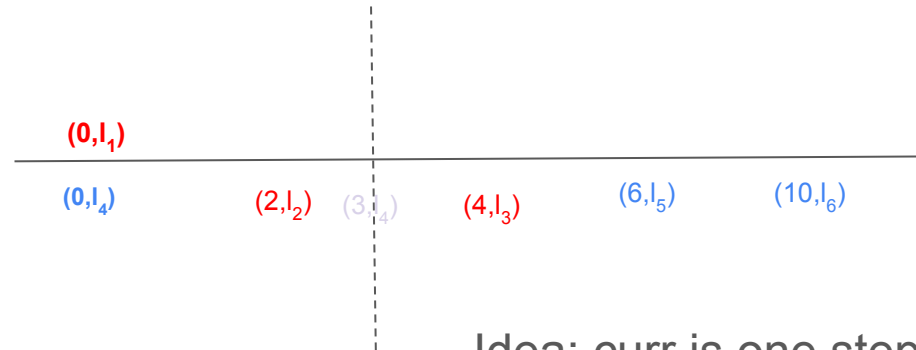
for curr = 2,...,10

Curr = 3

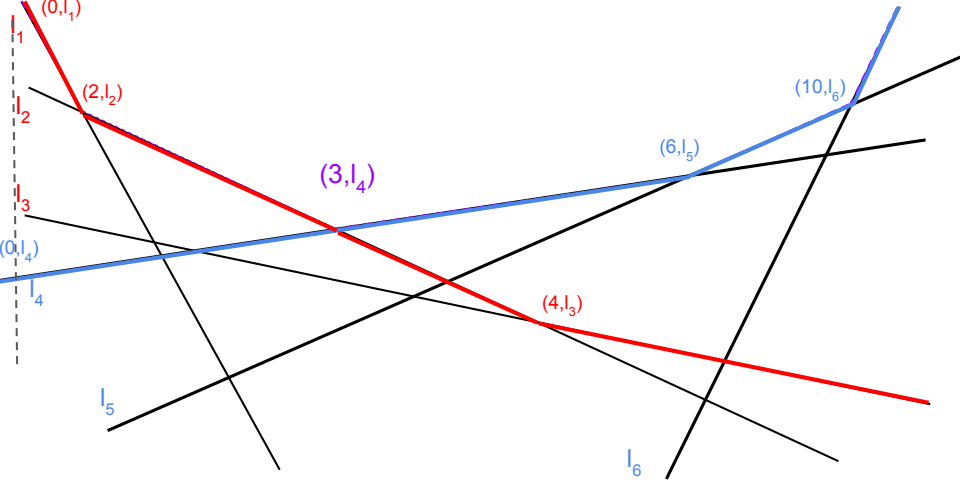
currRed: I_2

currBlue: I_4

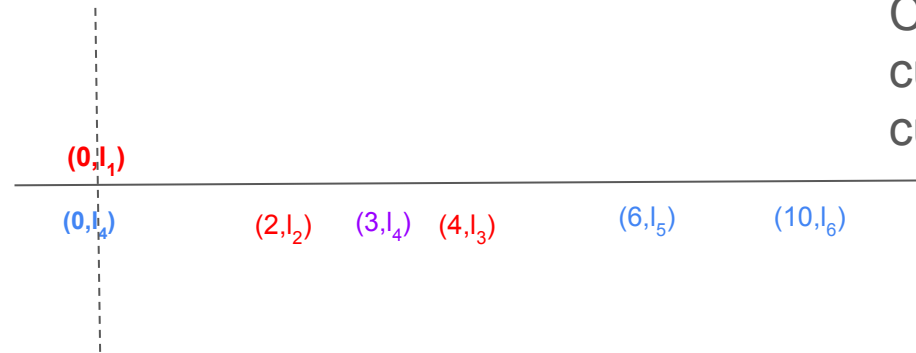
$(0, I_1)$



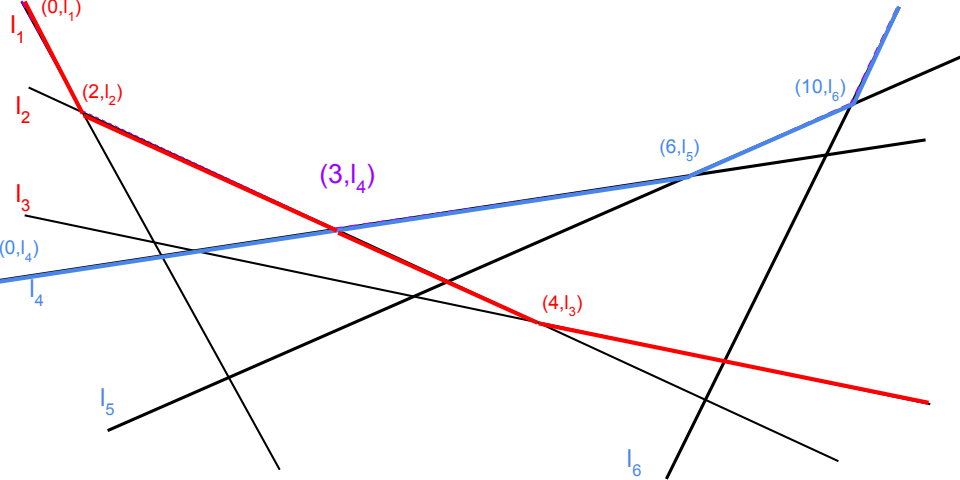
Idea: curr is one step ahead of whatever breakpt we want to add



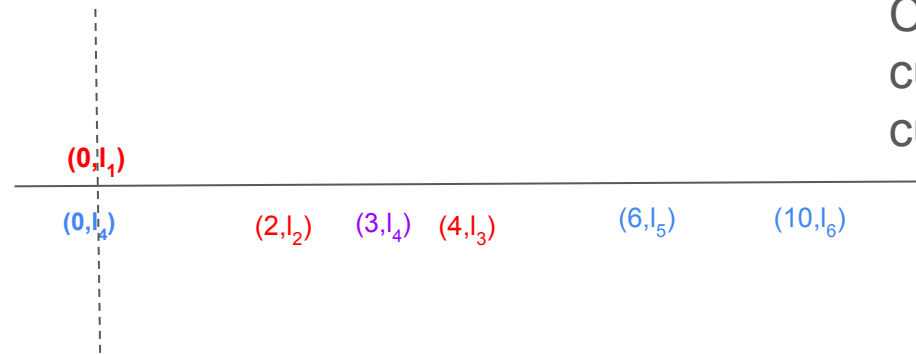
First Idea: Sweep a line left to right,
choose higher line based on curr breakpt



Curr = 0
currRed: I_1
currBlue: I_4

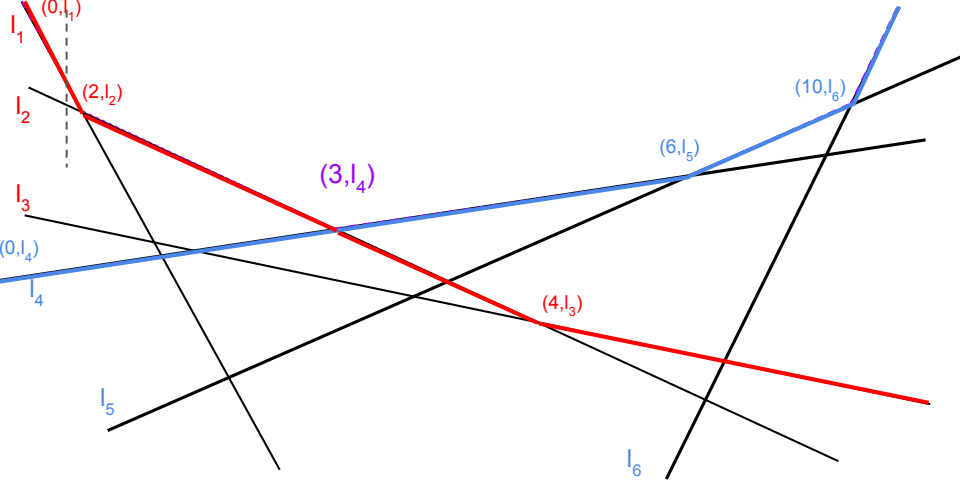


First Idea: Sweep a line left to right,
choose higher line based on curr breakpoint



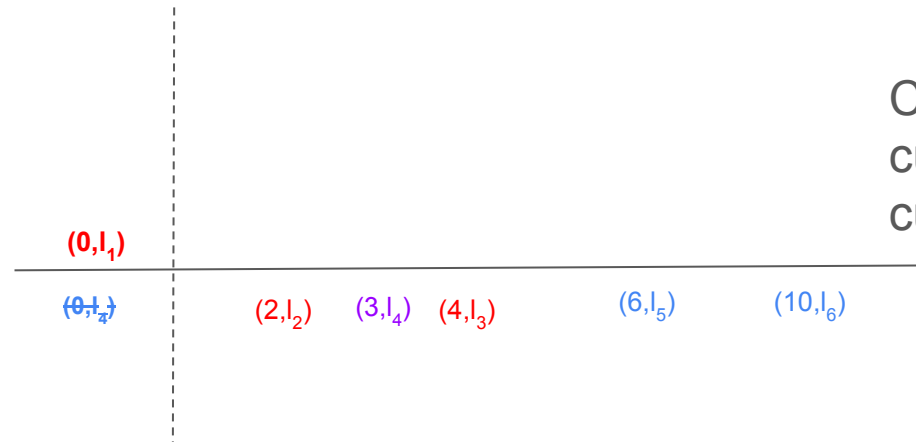
Curr = 0
currRed: I_1
currBlue: I_4

$(0, I_1)$ added since $I_1(0) > I_4(0)$



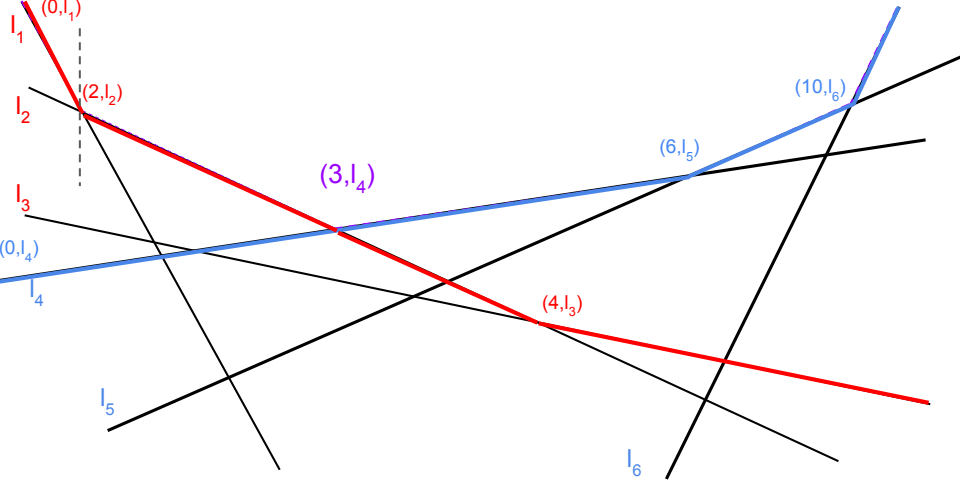
First Idea: Sweep a line left to right,
choose higher line based on curr breakpt

Still on the same line so
nothing happens for curr = 1



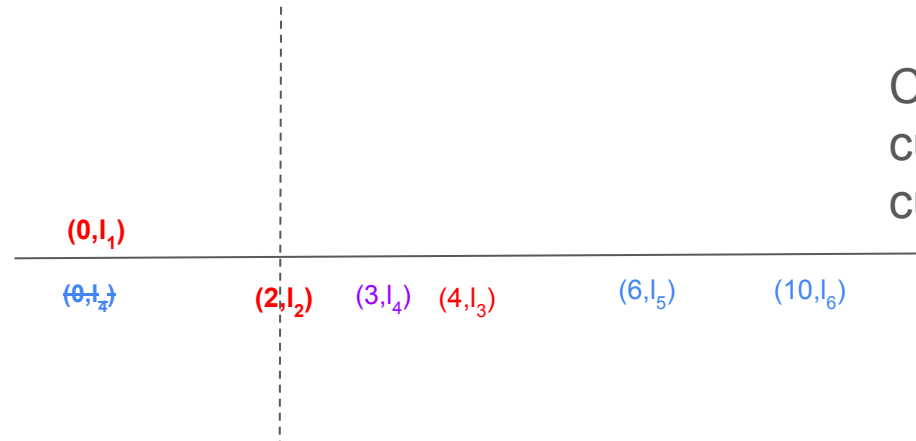
Curr = 1
currRed: I_1
currBlue: I_4

$(0, I_1)$



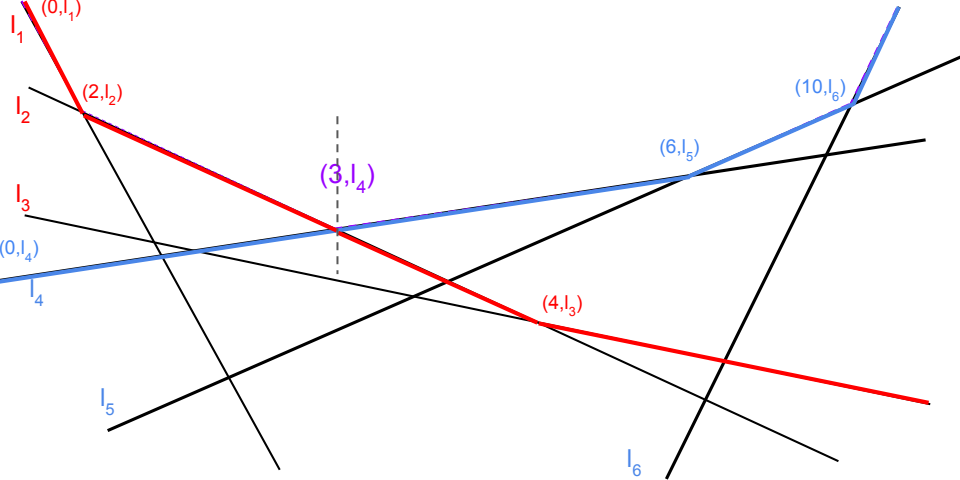
First Idea: Sweep a line left to right,
choose higher line based on curr breakpt

We are at a new line and breakpt I_2

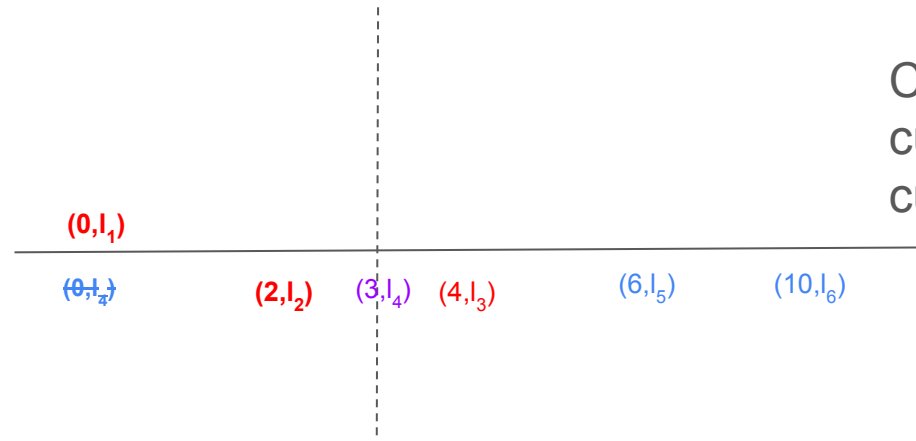


Curr = 2
currRed: I_2
currBlue: I_4

$(0, I_1)$ $(2, I_2)$



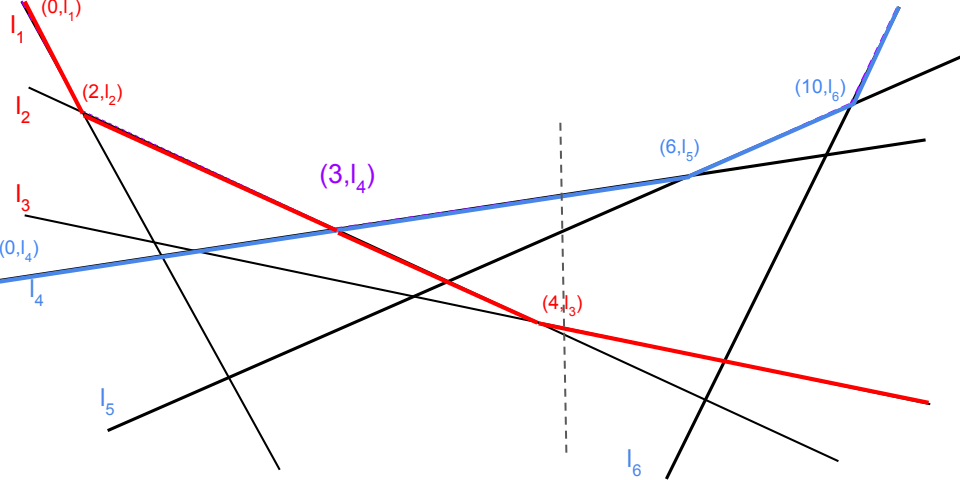
First Idea: Sweep a line left to right,
choose higher line based on curr breakpoint



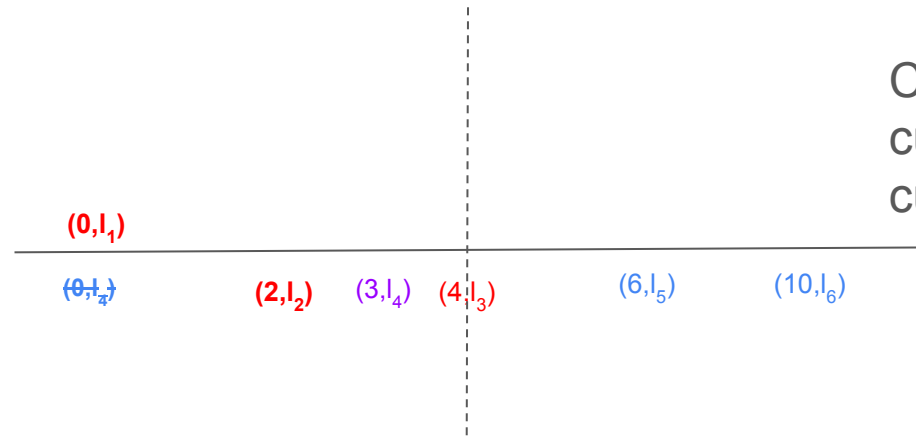
Curr = 3
currRed: I_2
currBlue: I_4

(0, I_1) (2, I_2)

Hmm we miss pt 3 for now..



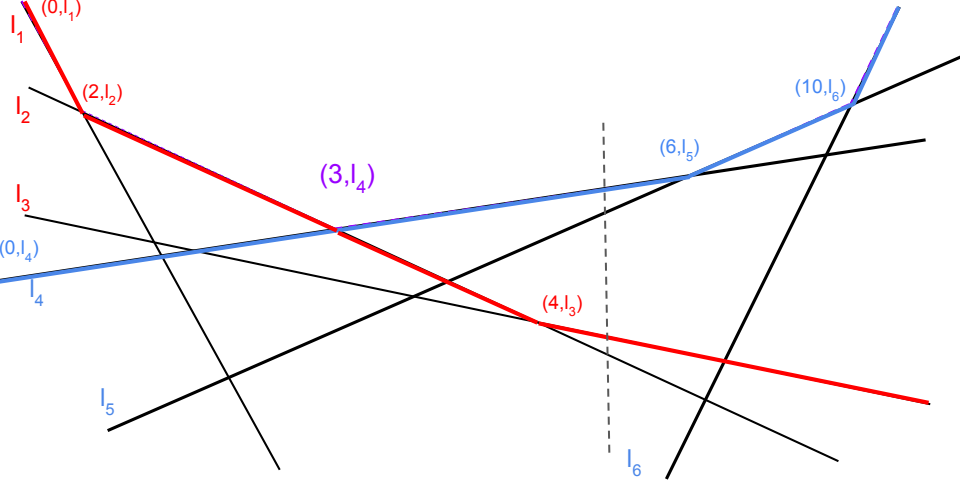
First Idea: Sweep a line left to right,
choose higher line based on curr breakpoint



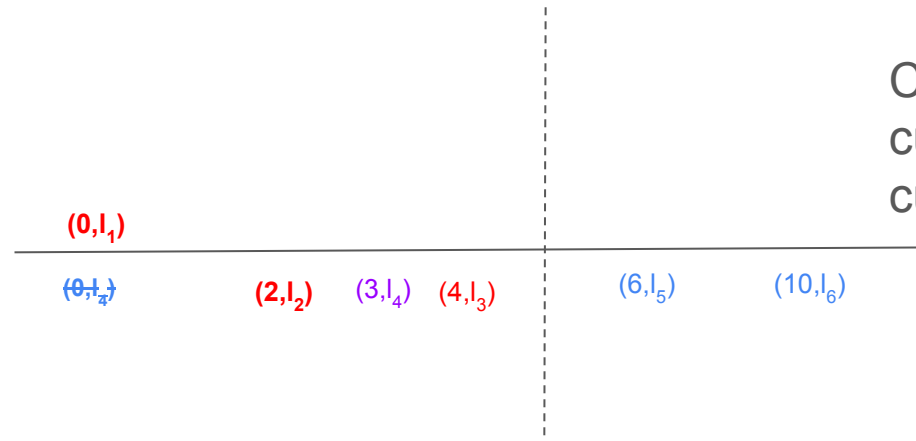
Curr = 4
currRed: I_3
currBlue: I_4

(0, I_1) (2, I_2) (4, I_3)

Hmm we miss pt 3 for now.. And add pt 4 incorrectly D:

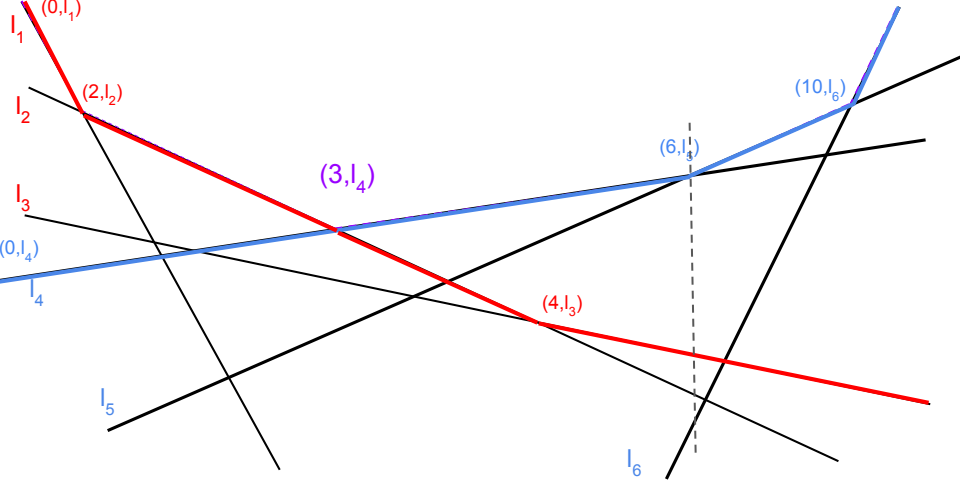


First Idea: Sweep a line left to right,
choose higher line based on curr breakpt

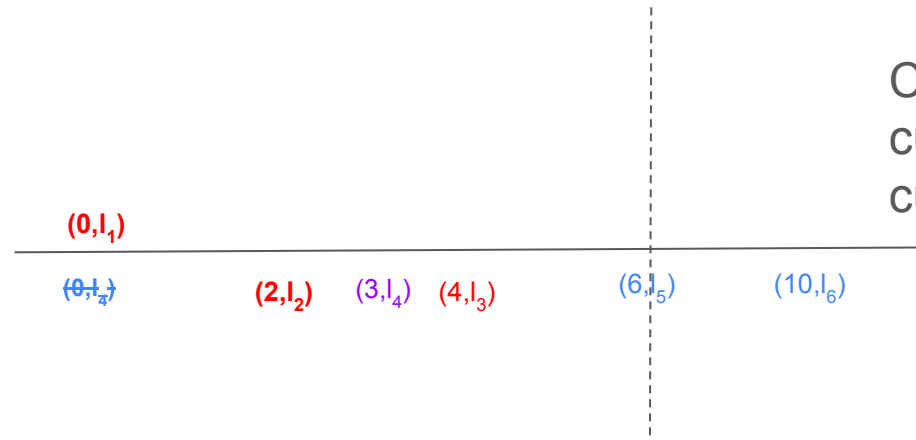


Curr = 5
currRed: I_3
currBlue: I_4

$(0, I_1)$ $(2, I_2)$ $(4, I_3)$



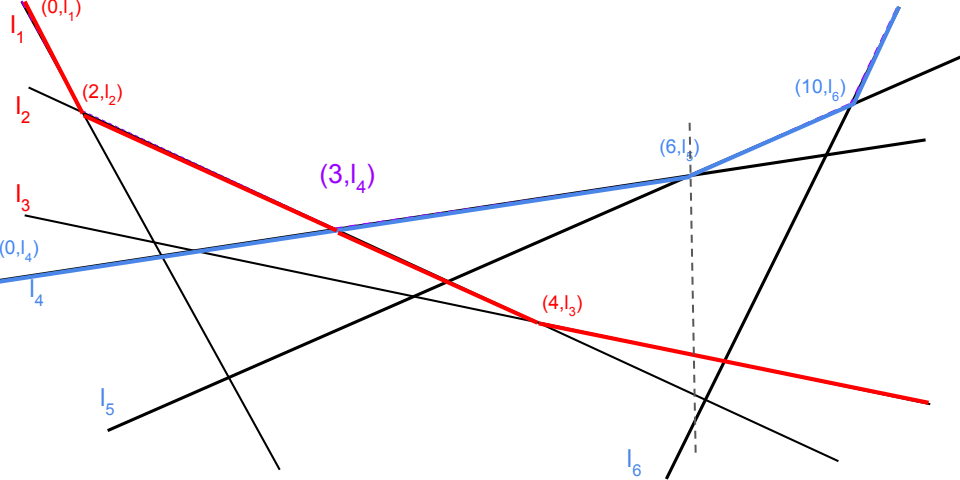
First Idea: Sweep a line left to right,
choose higher line based on curr breakpt



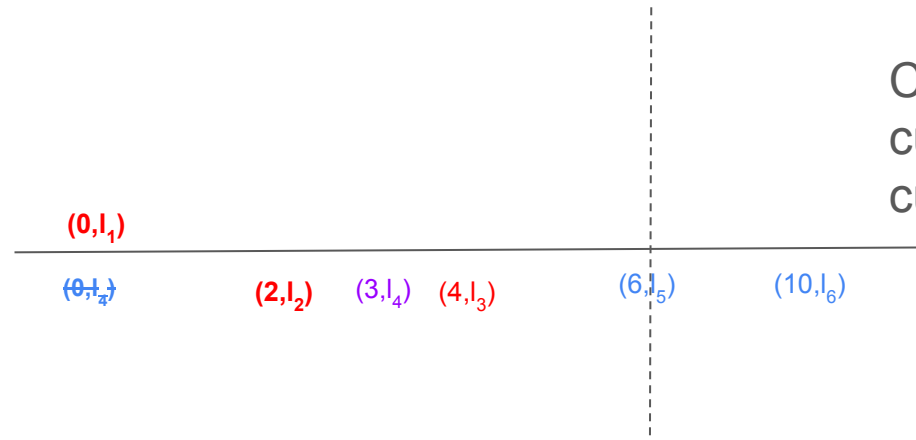
Curr = 6
currRed: I_3
currBlue: I_4

$(0, I_1)$ $(2, I_2)$ $(4, I_3)$ $(6, I_5)$

The rest is fine..



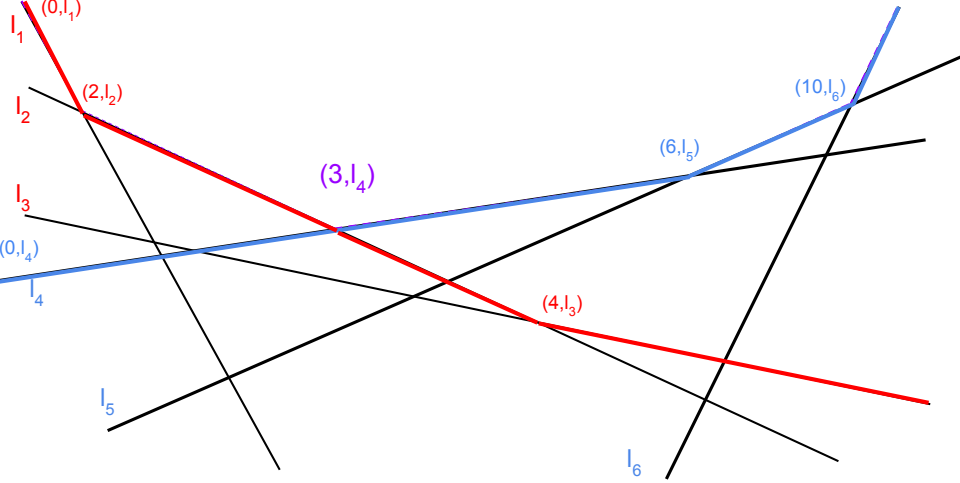
First Idea: Sweep a line left to right,
choose higher line based on curr breakpt



Curr = ..10
currRed: I_3
currBlue: I_4

$(0, I_1)$ $(2, I_2)$ $(4, I_3)$ $(6, I_5)$ $(10, I_6)$

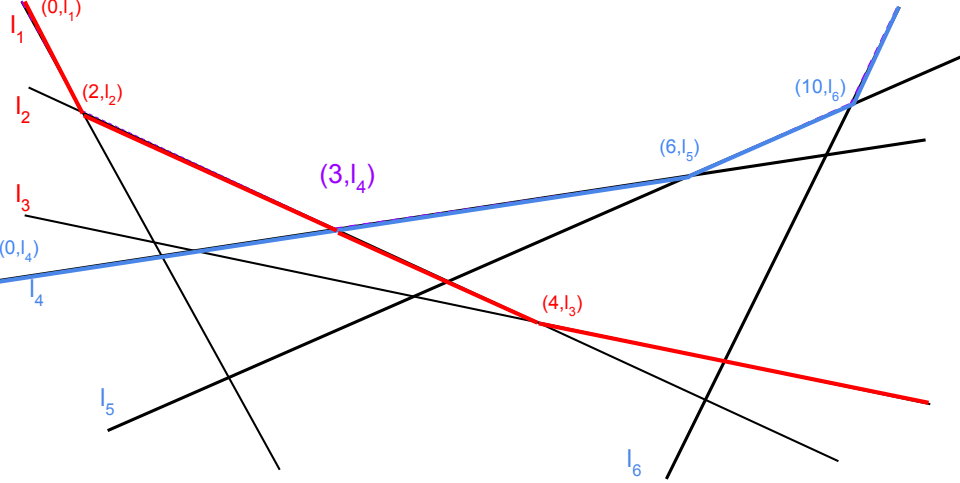
The rest is fine..



Sweep left to right by each break point
and add higher line

1. Order breakpoint/lines by x values $//O(n)$

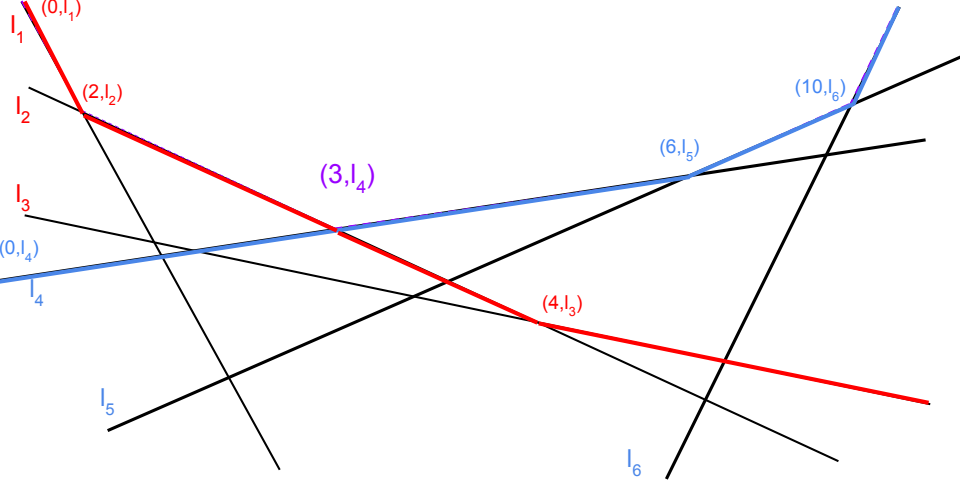
$(0, l_4)$ $(0, l_1)$ $(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$



Sweep left to right by each break point
and add higher line

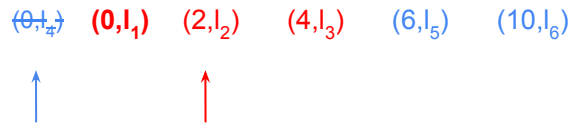
1. Order breakpoint/lines by x values //O(n)

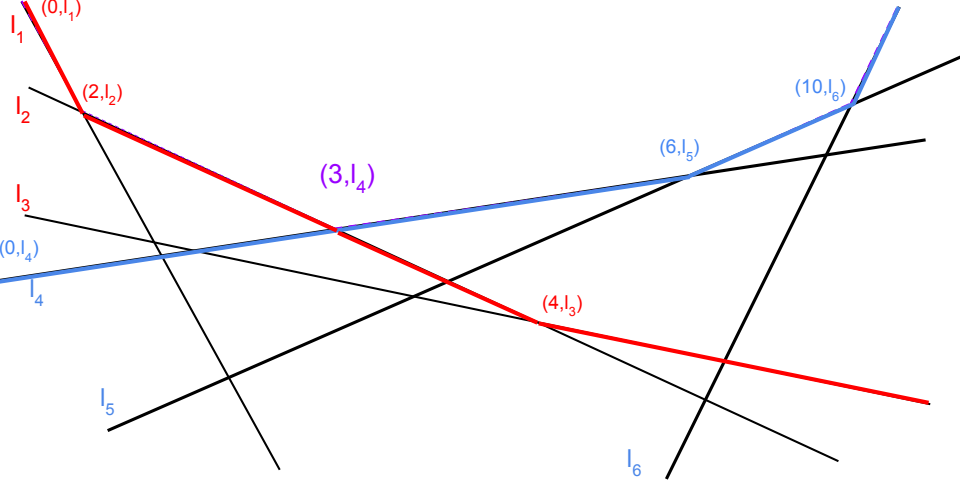
$(0, l_4)$ $(0, l_1)$ $(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $\uparrow$ $\uparrow$



Sweep left to right by each break point
and add higher line

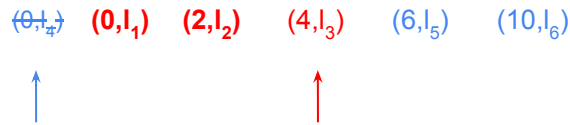
1. Order breakpoint/lines by x values //O(n)

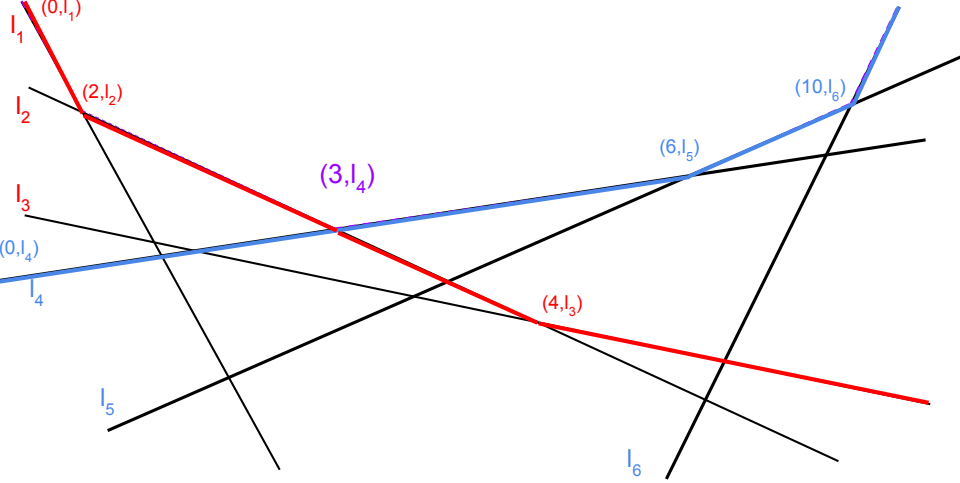




Sweep left to right by each break point
and add higher line

1. Order breakpoint/lines by x values //O(n)

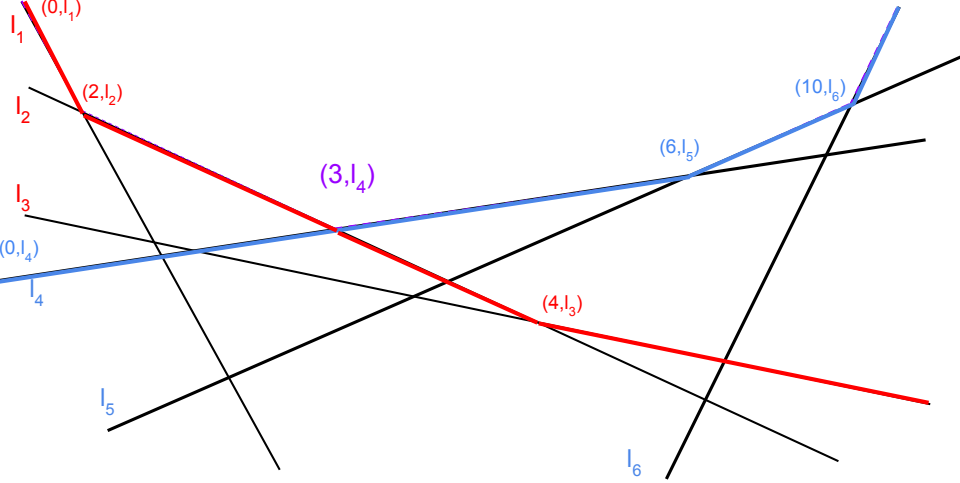




Sweep left to right by each break point
and add higher line

1. Order breakpoint/lines by x values //O(n)

~~(0, l4)~~ (0, l1) (2, l2) ~~(4, l3)~~ (6, l5) (10, l6)
 ↑ ↑
 (red) (blue)



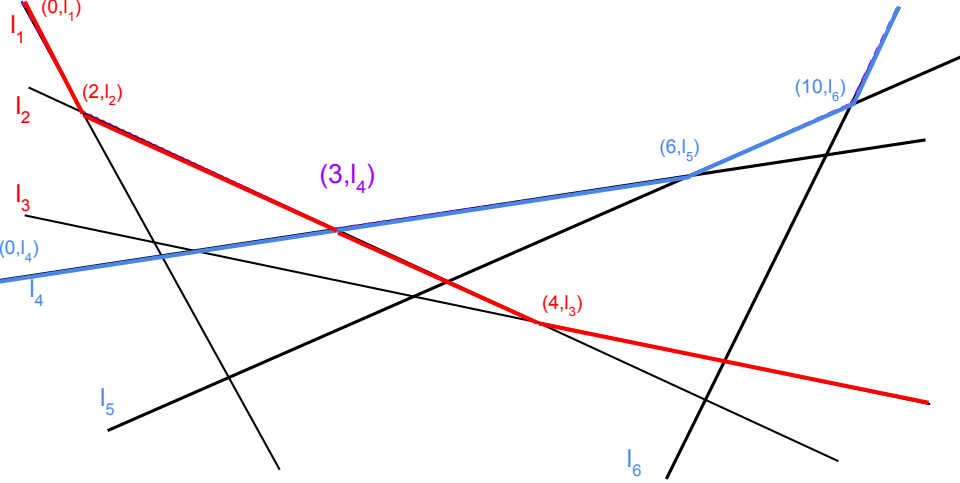
Sweep left to right by each break point and add higher line

1. Order breakpoint/lines by x values //O(n)

~~(0, l4)~~ (0, l1) (2, l2) ~~(4, l3)~~ (6, l5) (10, l6)

↑ ↑

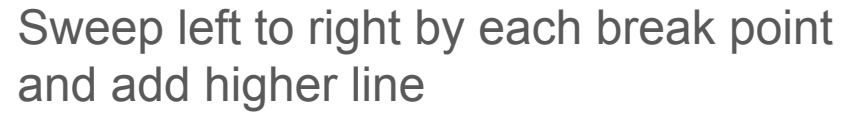
Missing (3, l4)?



Sweep left to right by each break point
and add higher line

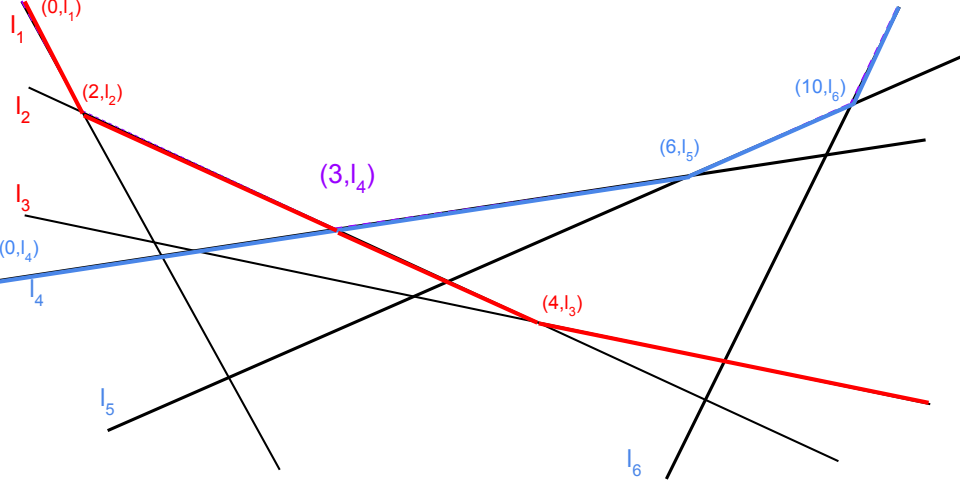
1. Order breakpoint/lines by x values //O(n)

$(0, l_4)$ $(0, l_1)$ $(2, l_2)$ $(4, l_3)$ $(6, l_5)$ $(10, l_6)$
 $\uparrow$ $\uparrow$



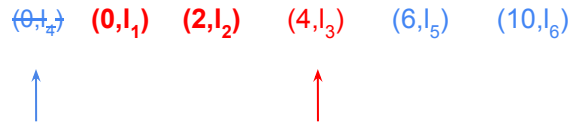
$(0, |_4)$ $(0, |_1)$ $(2, |_2)$ $(4, |_3)$ $(6, |_5)$ $(10, |_6)$

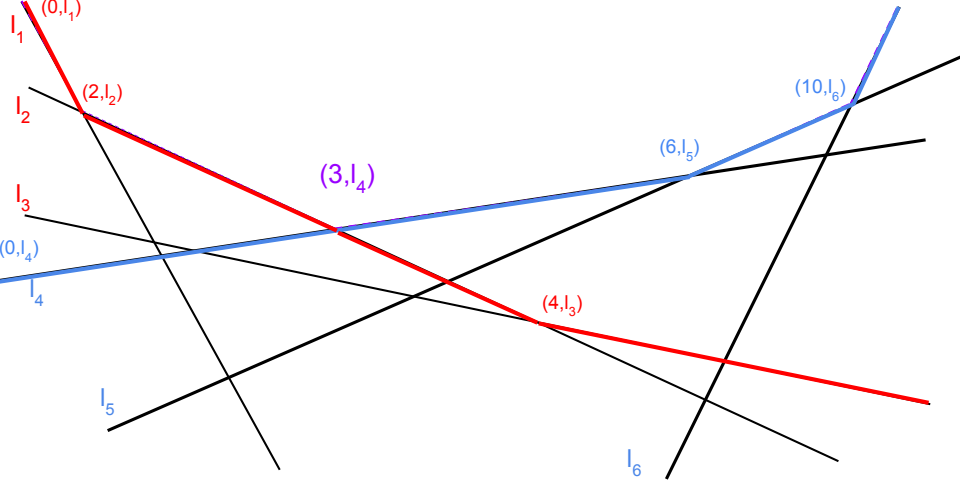
↑ ↑



Sweep left to right by each break point
and add higher line

1. Order breakpoint/lines by x values //O(n)



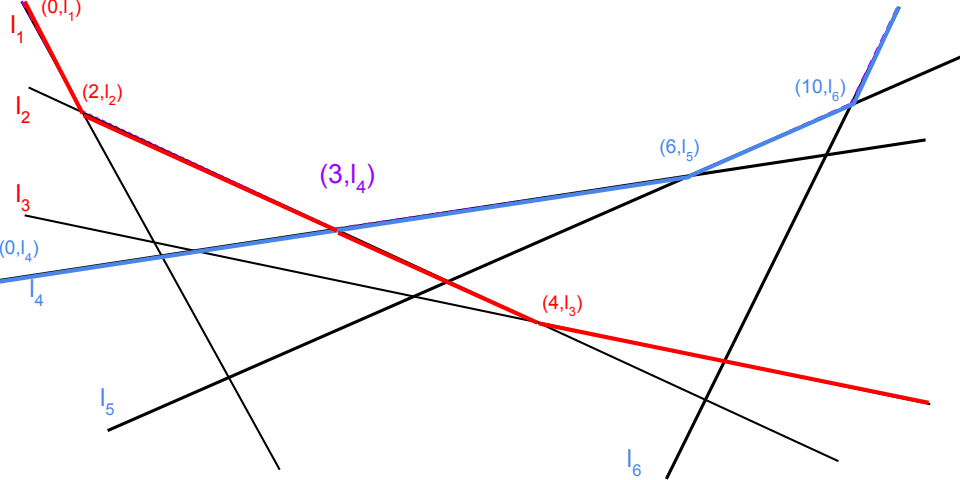


Sweep left to right by each break point and add higher line

1. Order breakpoint/lines by x values //O(n)

~~$(0, l_4)$~~ $(0, l_1)$ $(2, l_2)$ ~~$(4, l_3)$~~ $(6, l_5)$ $(10, l_6)$
 $\uparrow$ $\uparrow$

We switched between red and blue here!
There is some intersection pt between l_2 and l_5



Sweep left to right by each break point and add higher line

We switched between red and blue here!
There is some intersection pt between l_2 and l_5

1. Order breakpoint/lines by x values //O(n)

