1. ja 2. detsembril on loeng ja praktikum ära vahetatud

Antud punktid $p_1, \ldots, p_n$ (kõik erinevad). Leida nende seast kaks teineteisele lähimat punkti.

Lihtne lahendus — vaatame kõik punktipaarid läbi, leiame nendevaheliste kauguste seast minimaalse. Keerukus: $\Theta(n^2)$.

Punktide $p_1 = (x_1, y_1)$ ja $p_2 = (x_2, y_2)$ vaheline kaugus $d(p_1, p_2)$ on

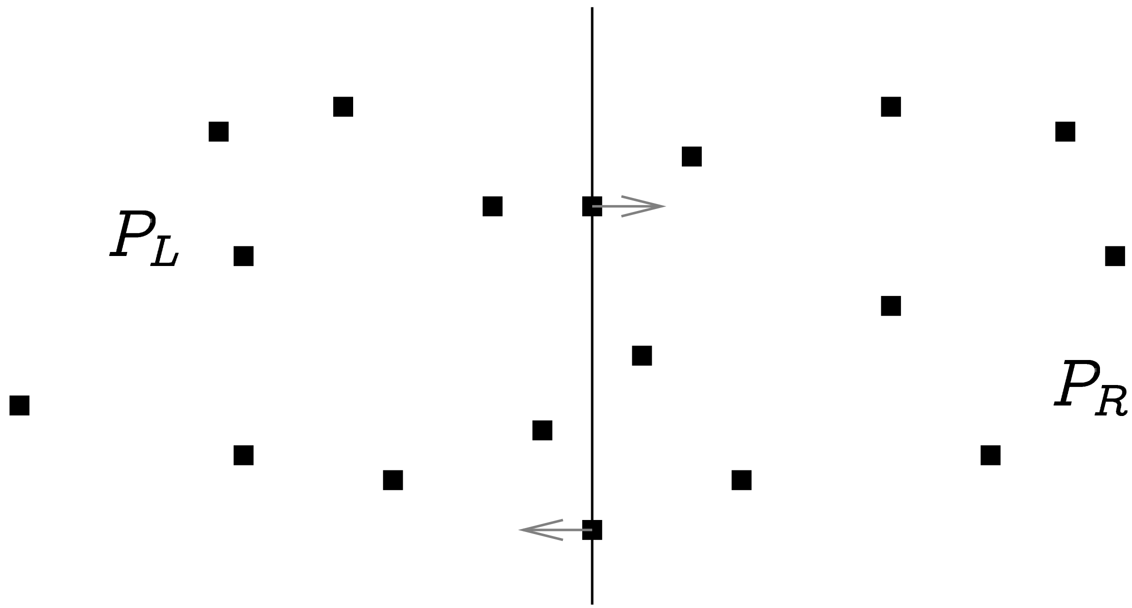
$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \ .$$

Kui me vaid kaugusi võrrelda tahame, siis piisab kauguste ruutude võrdlemisest. S.t. ruutjuurt pole tarvis võtta.

Rekursiivne lahendus:

1. Jaga punktihulk P kaheks võrdse suurusega osaks P_L ja P_R, leia kummastki osast vähima kaugusega punktipaar.
   - Rekursioon lõppeb, kui $|P| \leqslant 3$.
2. Vali neist kahest paarist lähem, olgu nendevaheline kaugus δ.
3. Kontrolli, kas leiduvad punktid $p_L \in P_L$ ja $p_R \in P_R$, nii et $d(p_L, p_R) \leqslant \delta$.
   - Piisab selliste punktide $p_L \in P_L$ vaatamisest, mis on hulgale P_R lähemal kui δ. Sama $p_R \in P_R$ jaoks...

Jaotamise teeme vertikaalse sirgega.

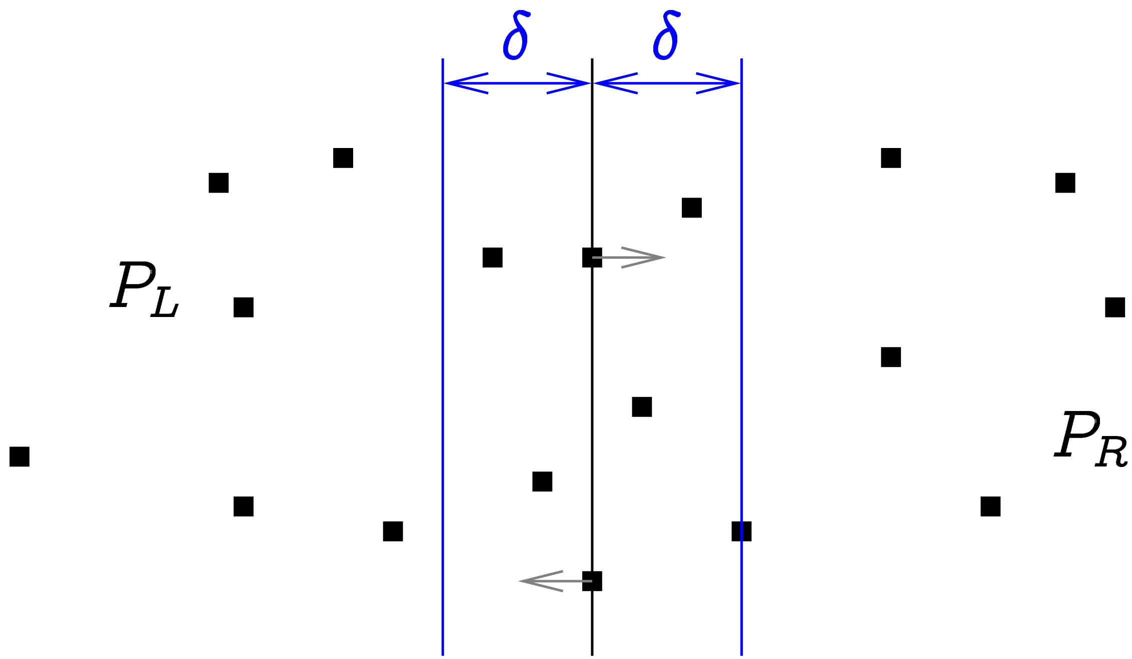


Sirgele jäävad punktid võivad kuuluda nii ühte kui ka teise poolde.

Olgu $X_1, \ldots, X_n$ punktid $p_1, \ldots, p_n$, sorteeritud x-koordinaadi järgi. Massiivi X kasutame vertikaalse sirge valikul.

Olgu δ_L vähima kaugusega punktipaari kaugus osas P_L ja δ_R sama asi osas P_R. Olgu $\delta = \min(\delta_L, \delta_R)$.

Kui $p_L \in P_L$ ja $p_R \in P_R$ on sellised, et $d(p_L, p_R) < \delta$, siis on p_L ja p_R eraldavast vertikaalsest joonest kaugusel $\leqslant \delta$.

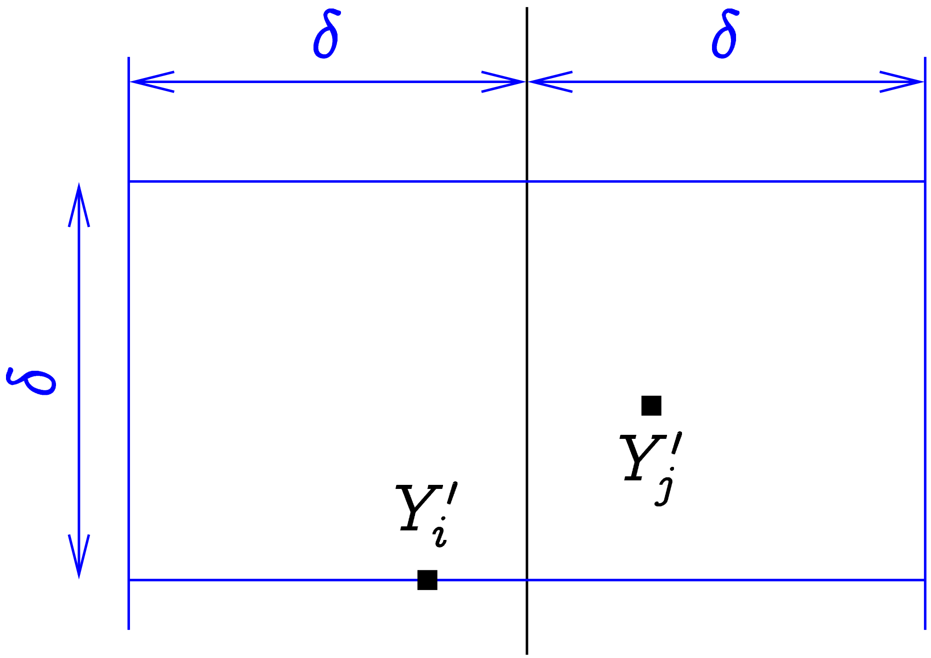


Olgu $P' \subseteq P$ kõigi selliste punktide hulk, mis on valitud vertikaalsest sirgest kaugusel $\leqslant \delta$. Olgu Y' nendesamade punktide järjend, y-koordinaadi järgi sorteeritud.

Olgu Y'_i ja Y'_j $(j > i)$ teineteisele lähimad punktid P'-st. Kui $d(Y'_i, Y'_j) \leqslant \delta$, siis $j - i \leqslant 6$.

S.t. (kui eelmine lõik on õige, siis) selleks, et leida teineteisele lähim punktipaar P'-st tuleb vaadata läbi massiiv Y' ja leida mingi punkti Y'_i kaugused ainult punktidest $Y'_{i+1}, \ldots, Y'_{i+6}$. S.t. teineteisele lähima punktipaari leidmine on tehtav lineaarses ajas.

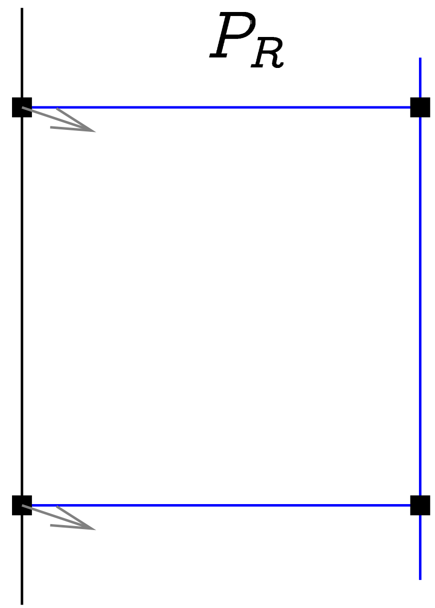
Kui $d(Y_i', Y_j') \leqslant \delta$, siis asuvad punktid Y_i' ja Y_j' ristkülikus mõõtmetega $2\delta \times \delta$.



Üldisust kitsendamata loeme, et $Y_i' \in P_L$. Vaatame juhtu, kus Y_i' ei asu vertikaalsel sirgel, mis eraldab P_L-i ja P_R-i.

Vaatame kõiki P_R-i kuuluvaid punkte, mis sinna ristkülikusse kuuluvad. Iga kahe P_R-i kuuluva punkti vaheline kaugus on vähemalt δ.

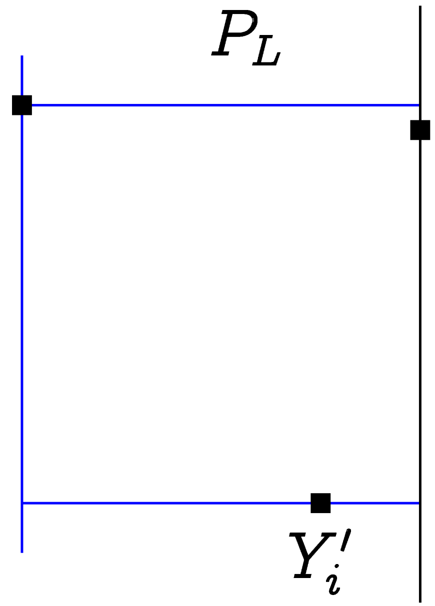
Nad kõik kuuluvad ruutu mõõtmetega $\delta \times \delta$.



Mahub ülimalt neli tükki.

Vaatame kõiki P_L-i kuuluvaid punkte, mis sinna ristkülikusse kuuluvad. Iga kahe P_L-i kuuluva punkti vaheline kaugus on vähemalt δ.

Nad kõik kuuluvad ruutu mõõtmetega $\delta \times \delta$.



Mahub ülimalt kolm tükki (kaasa arvatud Y_i').

Kui Y_i' asuks P_L-i ja P_R-i eraldaval vertikaalsel sirgel, siis mahuks vasakule poole neli punkti (k.a. Y_i') ja paremale poole kolm.

Seega kuulub vaadeldavasse ristkülikusse mõõtmetega $2\delta \times \delta$ ülimalt kuus punkti peale Y_i'.

Algoritm:

Eeltöö: sorteeri punktid $p_1, \ldots, p_n$ x-koordinaadi järgi massiivi X ja y-koordinaadi järgi massiivi Y. Kui kahel punktil on koordinaat võrdne, siis võta ettepoole see punkt, mille teine koordinaat on väiksem.

Põhitöö teeb ära funktsioon lähimad_punktid(X, Y, n).

lähimad_punktid(X, Y, n) on

if $n = 2$ **then return** (X_1, X_2)
if $n = 3$ **then**
    return lähim_kolmest(X_1, X_2, X_3)
$k := \lfloor n/2 \rfloor$
$X^L := X_{1 \ldots k}$; $X^R := X_{k+1 \ldots n}$
$l^L := 0$; $l^R := 0$
for $i := 1$ **to** n **do**
    if $Y_i.x < X_k.x$ **or** $Y_i.x = X_k.x \wedge Y_i.y \leqslant X_k.y$ **then**
        $l^L := l^L + 1$; $Y^L_{l^L} := Y_i$
    else
        $l^R := l^R + 1$; $Y^R_{l^R} := Y_i$
(q^L_1, q^L_2) := lähimad_punktid(X^L, Y^L, k)
(q^R_1, q^R_2) := lähimad_punktid($X^R, Y^R, n - k$)
$\ldots$

$\delta^{L2} := (q_1^L.x - q_2^L.x)^2 + (q_1^L.y - q_2^L.y)^2$

$\delta^{R2} := (q_1^R.x - q_2^R.x)^2 + (q_1^R.y - q_2^R.y)^2$

if $\delta^{L2} < \delta^{R2}$ **then**

    $\delta ruut := \delta^{L2};\ q_1 := q_1^L;\ q_2 := q_2^L$

else

    $\delta ruut := \delta^{R2};\ q_1 := q_1^R;\ q_2 := q_2^R$

$l' := 0$

for $i := 1$ **to** n **do**

    if $(Y_i.x - X_k.x)^2 \leqslant \delta ruut$ **then**

        $l' := l' + 1;\ Y'_{l'} := Y_i$

$\ldots$

for $i := 1$ **to** $l' - 1$ **do**
    for $j := 1$ **to** $\min(6, l' - i)$ **do**
        $druut := (Y_i'.x - Y_j'.x)^2 + (Y_i'.y - Y_j'.y)^2$
        if $druut < \delta ruut$ **then**
            $\delta ruut := druut$; $q_1 := Y_i'$; $q_2 := Y_j'$
return (q_1, q_2)

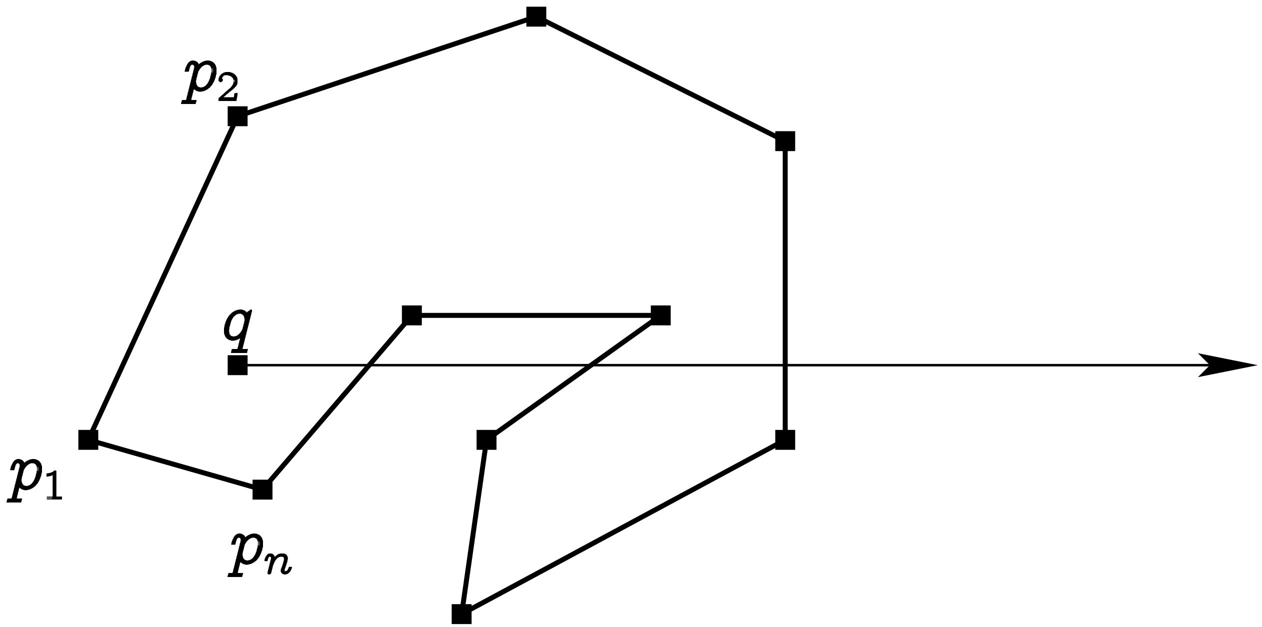
Keerukus:

- Eeltöö (sorteerimine) — $O(n \log n)$.
- Kui protseduuri lähimad_punktid tööaeg sisendil suurusega n on ülimalt $T(n)$, siis $T(n) = 2T(n/2) + O(n)$.
  - See $O(n)$ tuleb sellest, et meil on tsükleid üle kõigi punktide, pole aga kahekordseid tsükleid.
    - **for** $j := 1$ **to** $\min(6, l' - i)$ **do** ei loe.

  See rekurrents on lahendatav esimeses loengus tõestatud teoreemiga. Saame $T(n) = O(n \log n)$.

Kokku seega $O(n \log n)$.

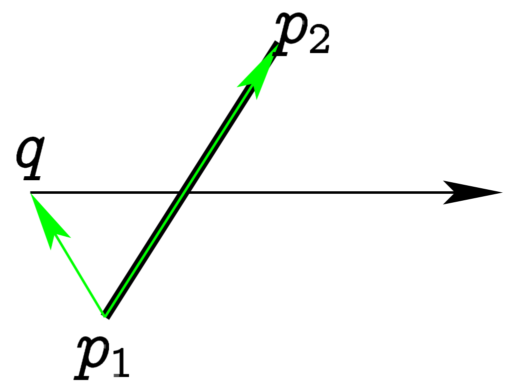
Olgu antud hulknurk tippudega $p_1, \ldots, p_n$ ning punkt q. Kas q asub hulknurga $p_1 p_2 \cdots p_n$ sees?

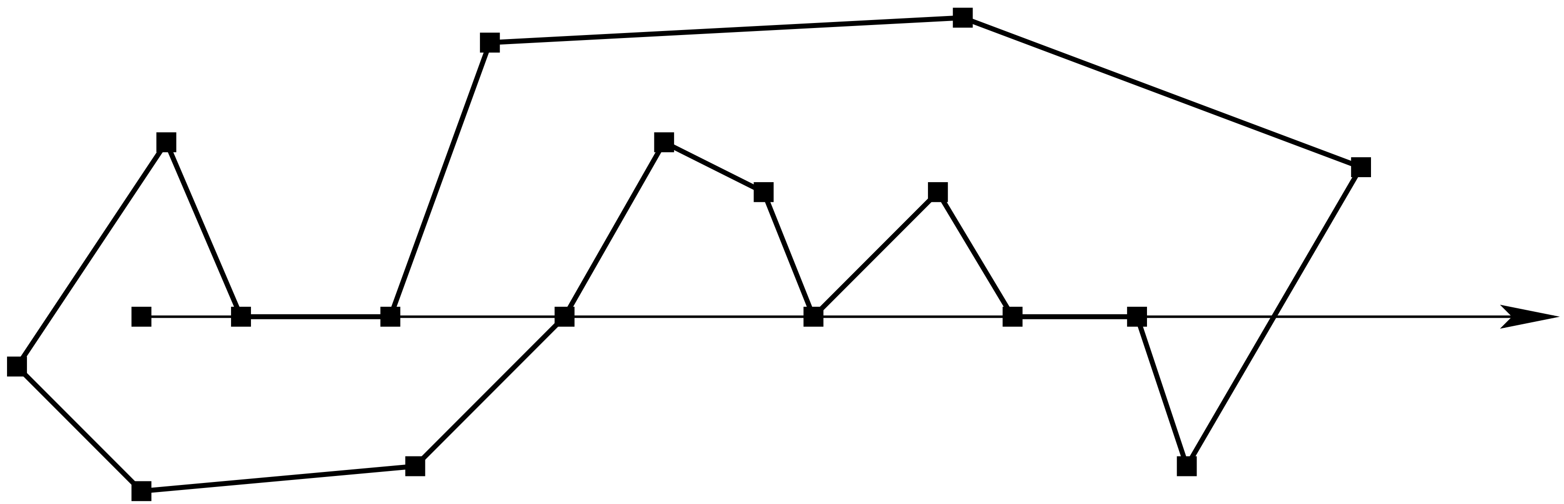


Vaatame mingit kiirt, mis lähtub punktist q. Kui q asub hulknurga sees, siis lõikab see kiir seda hulknurka paaritu arv kordi, muidu paarisarv kordi.

Millal lõikab lõik $\overline{p_1 p_2}$ (loeme, et $p_1.y \leqslant p_2.y$) horisontaalset kiirt, mis läheb punktist q paremale?

- $p_1.y \leqslant q.y \leqslant p_2.y$;
- $\overrightarrow{p_1 q}$ jääb $\overrightarrow{p_1 p_2}$-st vasakule.





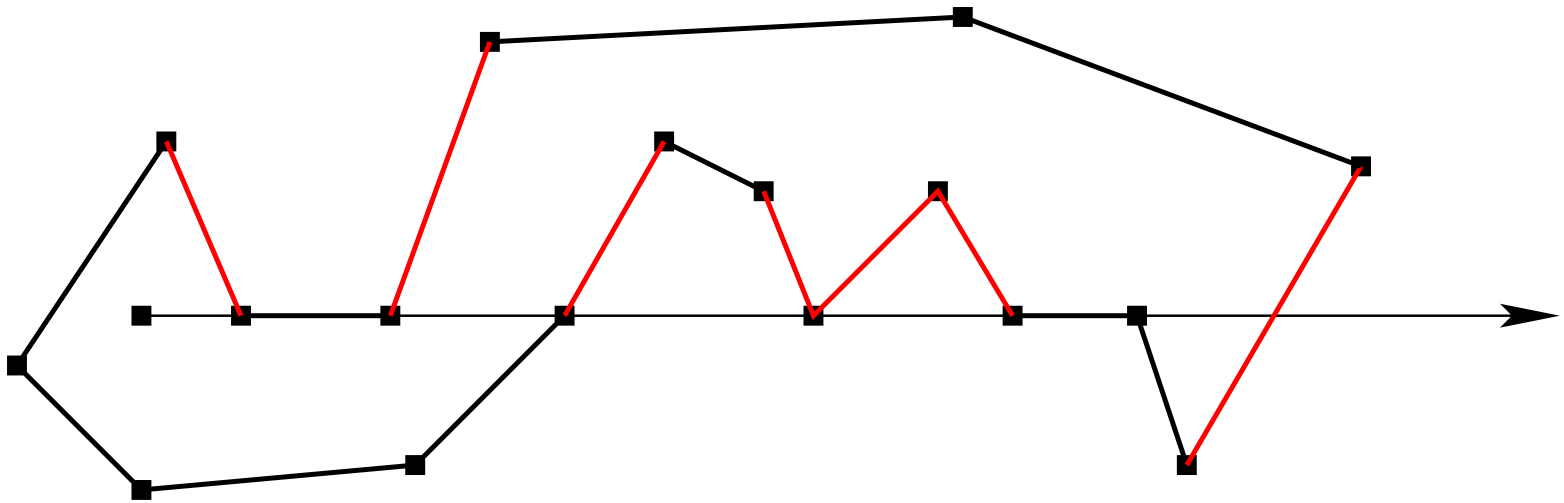
Mida teha siis, kui külje $\overline{p_1 p_2}$ (kus $p_1.x \leqslant p_2.x$) mõni tipp on q-st algaval kiirel?

Loeme nii:

- Kui $p_2.x = q.x$, siis ei lõiku.
- Kui $p_1.x = q.x < p_2.x$, siis lõikub.

S.t. kiir on „kaduvväikesel määral ülevalpool“ oma tegelikust asukohast.

Näide:

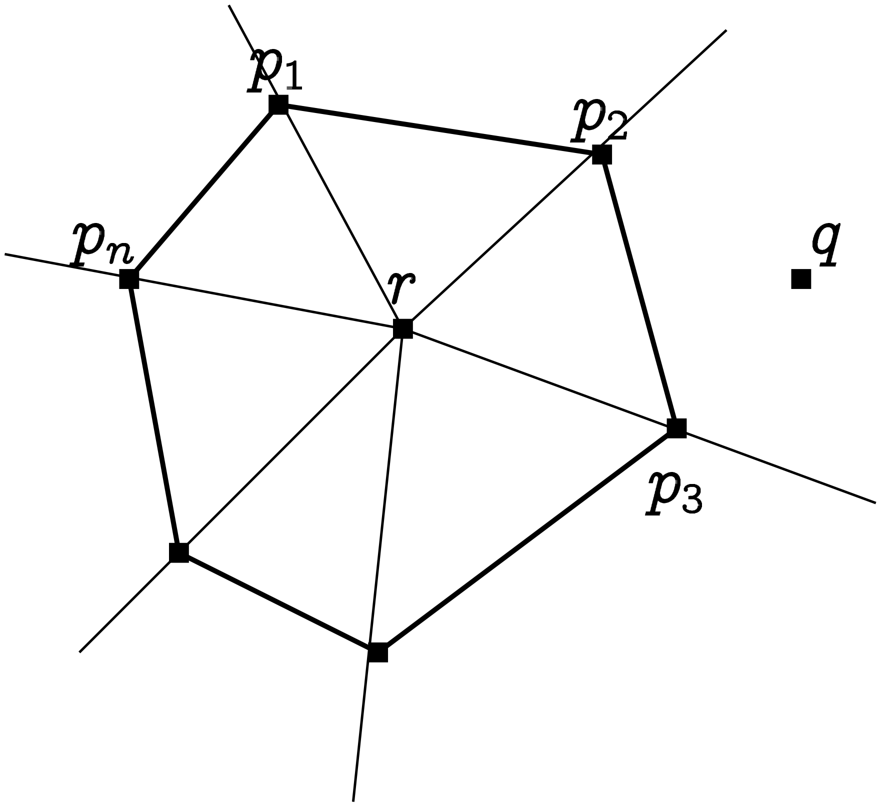


Seitse lõikumist $\Rightarrow$ sees.

Kõdunud juhu muutsime mittekõdunuks väikese häirituse lisamisega.

Keerukus — $O(n)$.

Kumera hulknurga puhul on punkti kuulumist hulknurka võimalik välja selgitada kiiremini, ajaga $O(\log n)$. Loeme, et $p_0, \ldots, p_1$ on antud kellaosuti liikumise suunas.



1. Olgu r mingi punkt hulknurga sisemuses. Sobib suvalise kolme tipu keskmine.

2. Leia, millisesse sektorisse $p_i r p_{i+1}$ kuulub punkt q. See on tehtav kahendotsimisega.

3. Leia, kas $\overline{rq}$ ja $\overline{p_i p_{i+1}}$ lõikuvad.

Olgu antud hulknurk $p_1 p_2 \cdots p_n$, tipud olgu kellaosuti liikumise suunas.

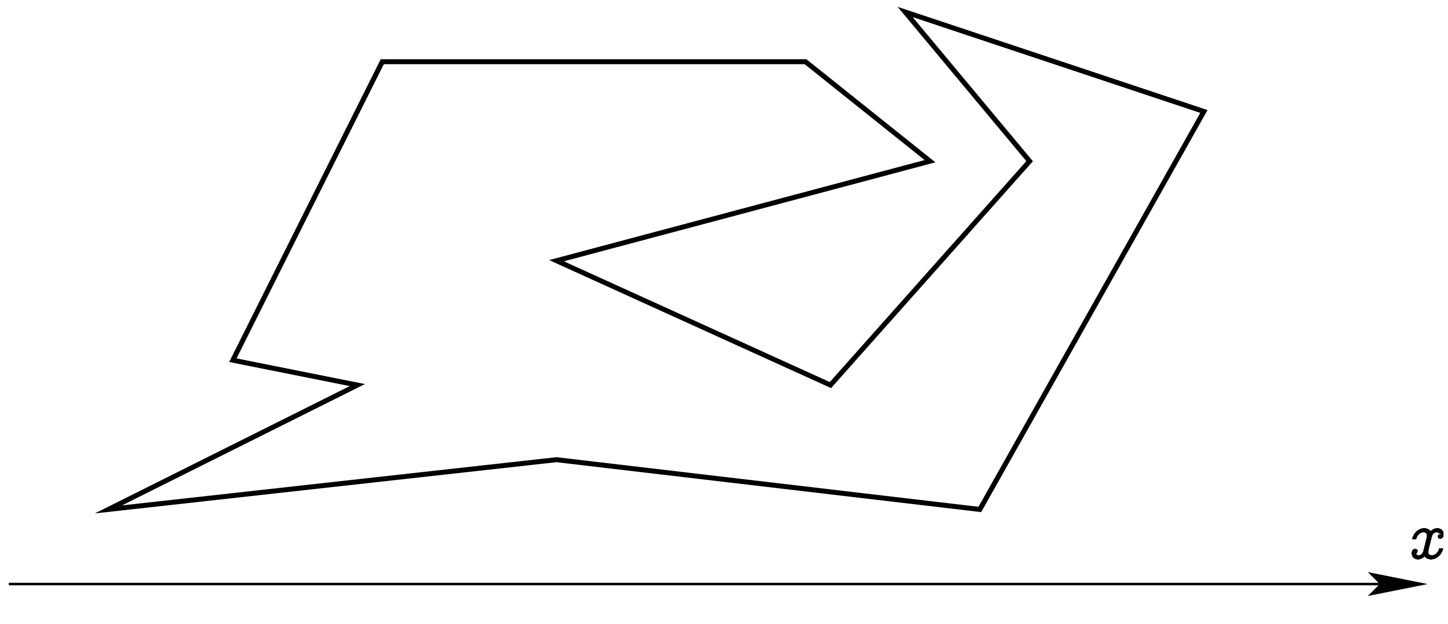
Kuidas leida selle hulknurga pindala?

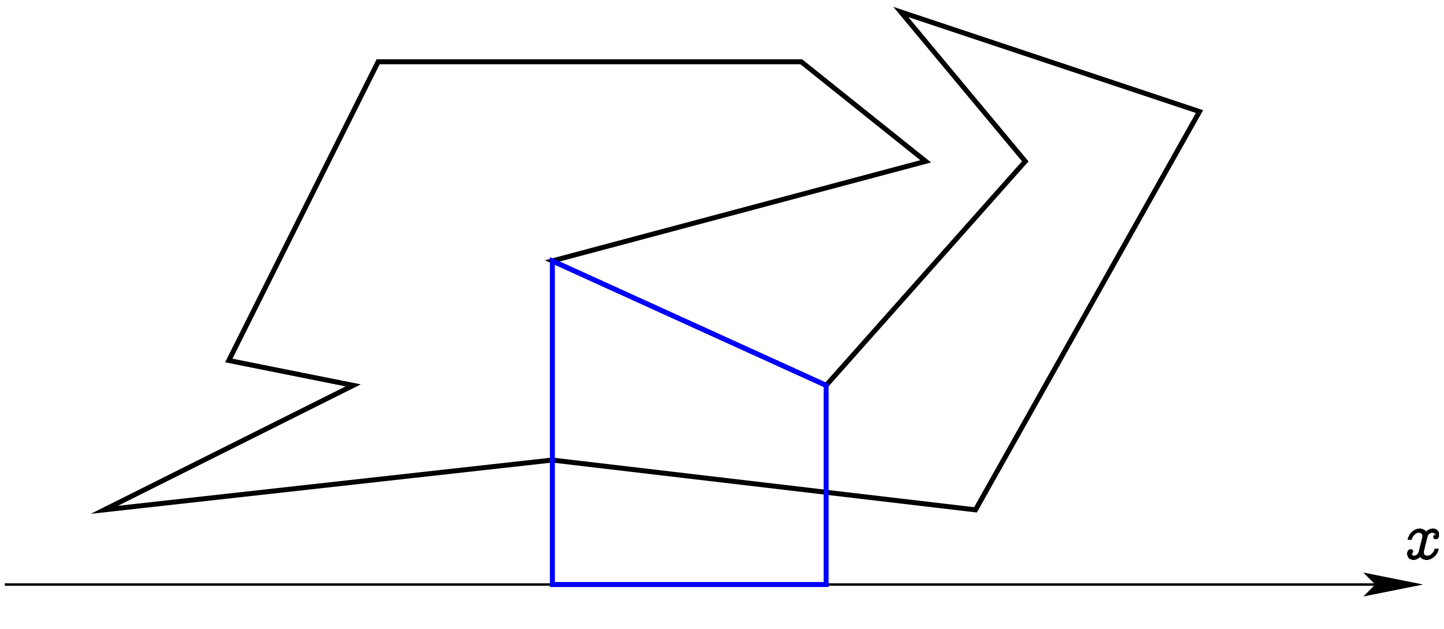
Olgu $p_i = (x_i, y_i)$. Loeme, et $y_i \geqslant 0$. Leiame trapetsite

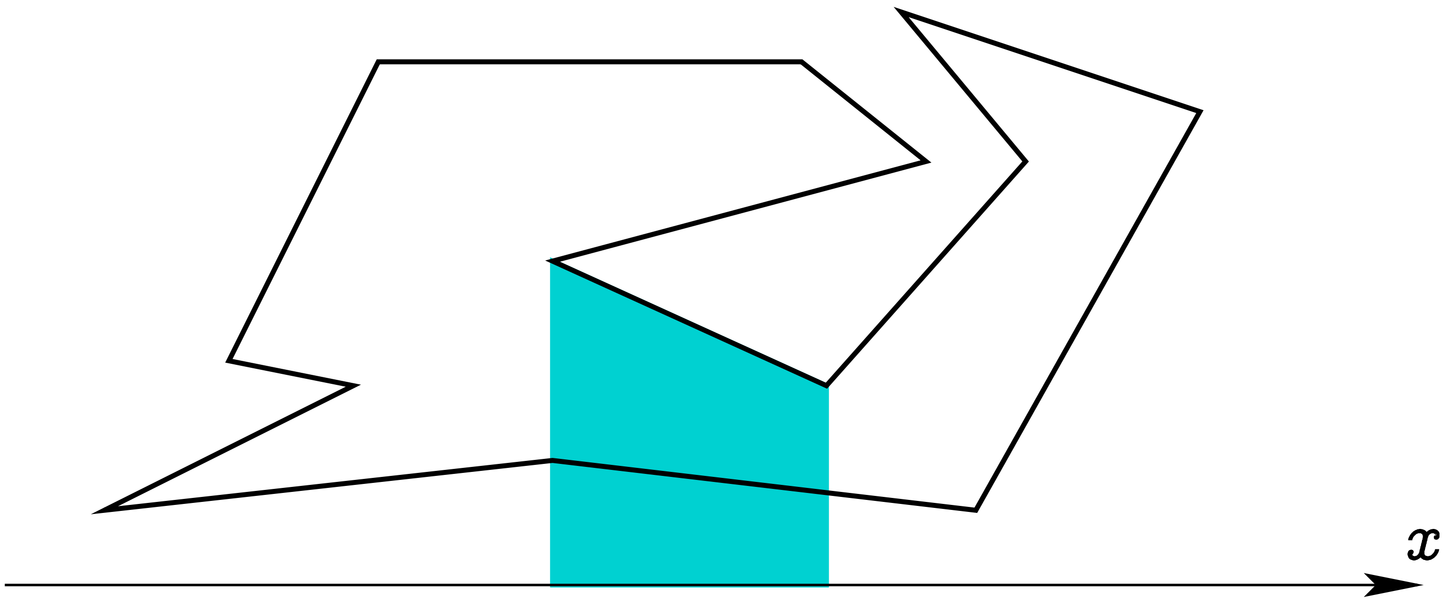
$$(x_i, 0)—(x_i, y_i)—(x_{i+1}, y_{i+1})—(x_{i+1}, 0)—(x_i, 0)$$

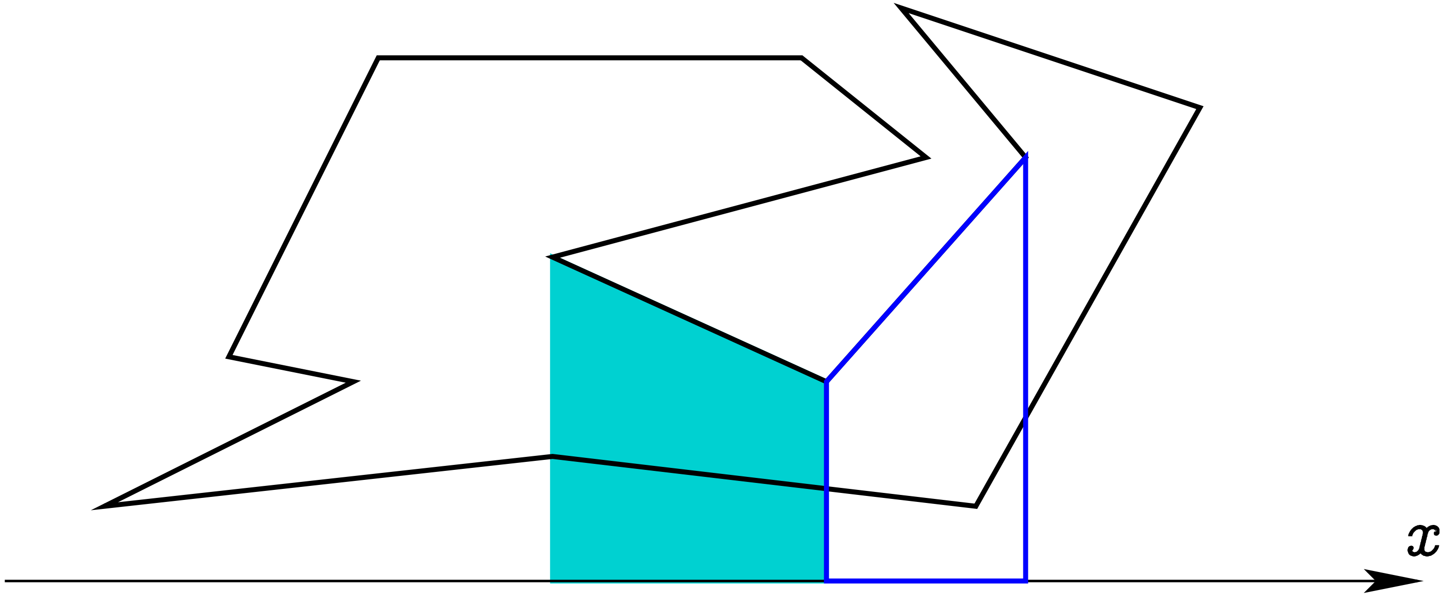
pindalad. Siin $1 \leqslant i \leqslant n$ ja $i+1$ on leitud mod n. Pindalad on *märgiga* — kui $x_{i+1} > x_i$, siis on pindala positiivne, muidu negatiivne.

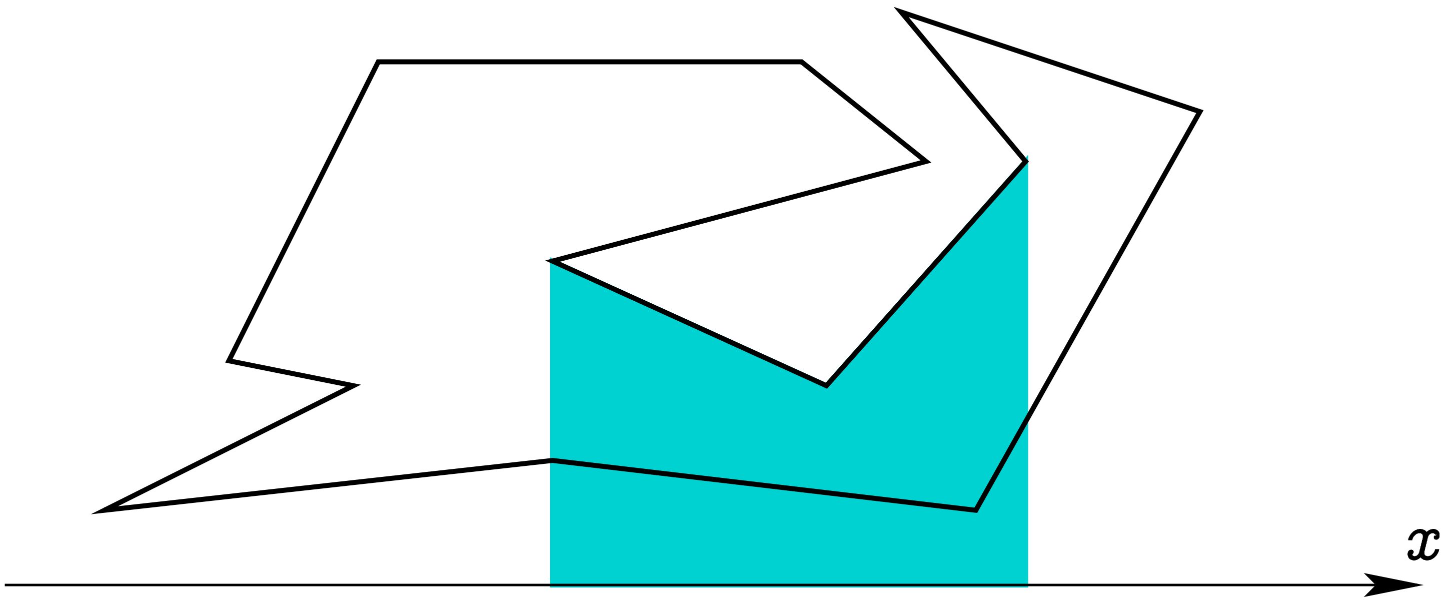
Trapetsite pindalate summa on hulknurga pindala.

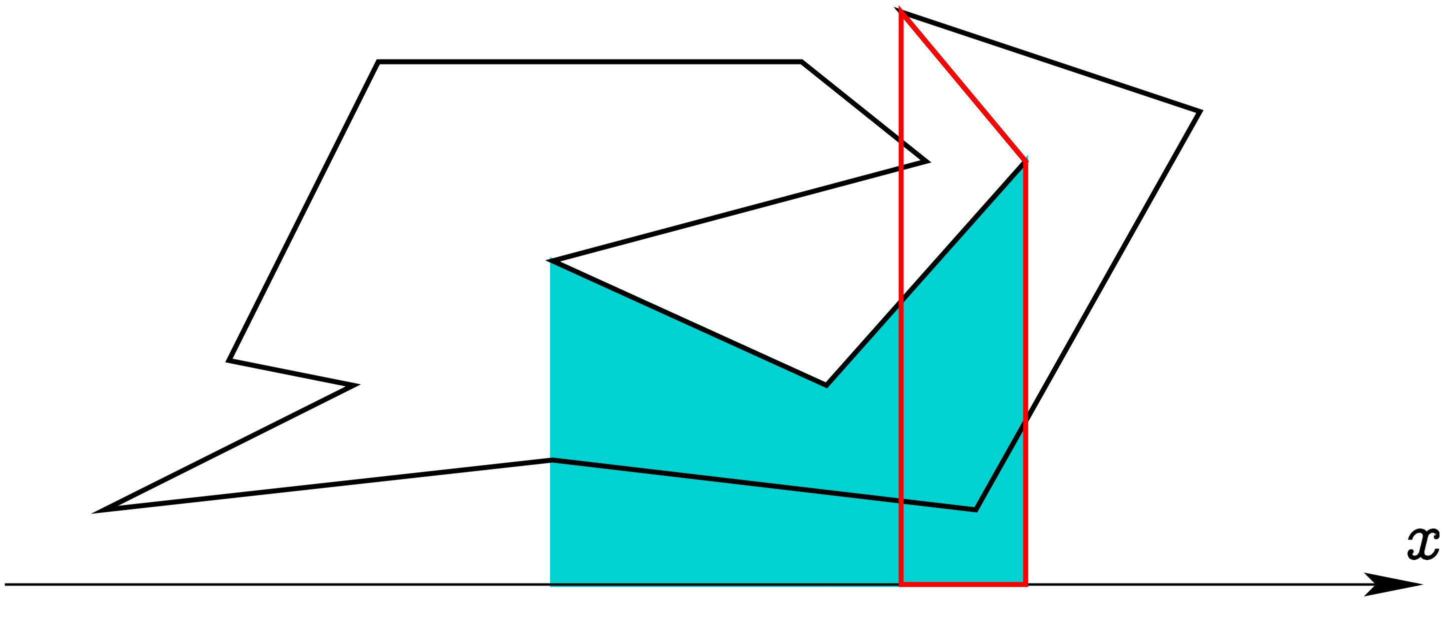

x

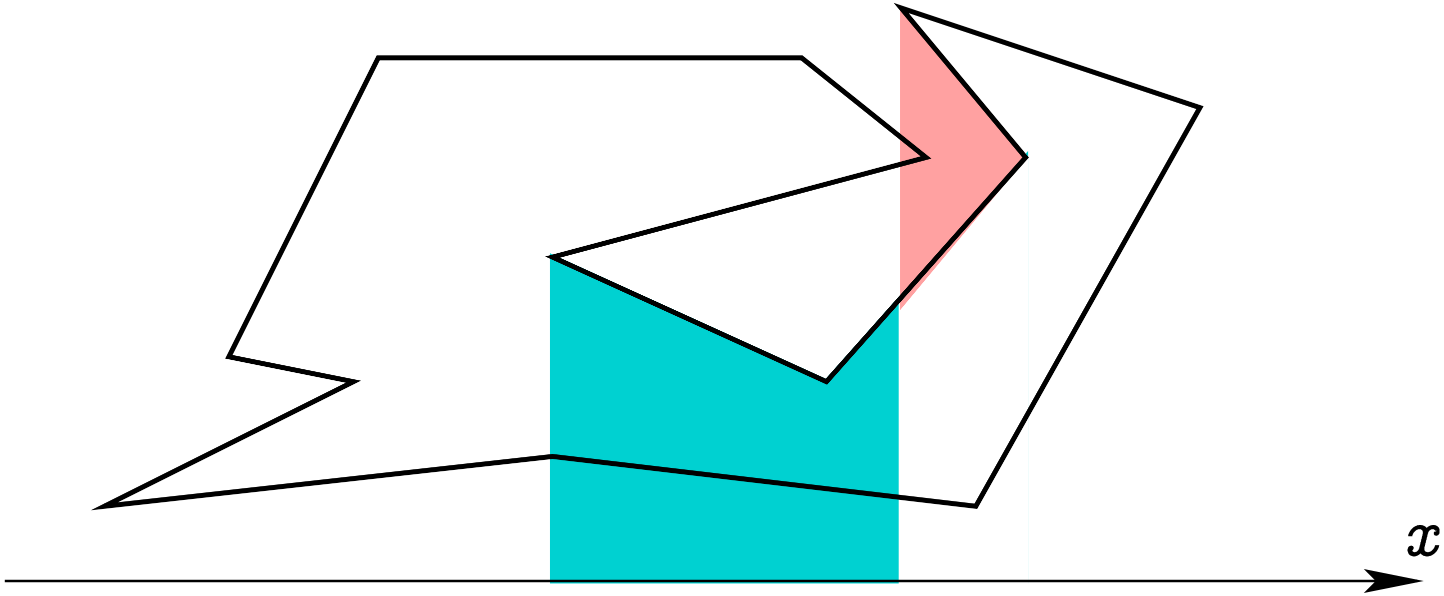

x

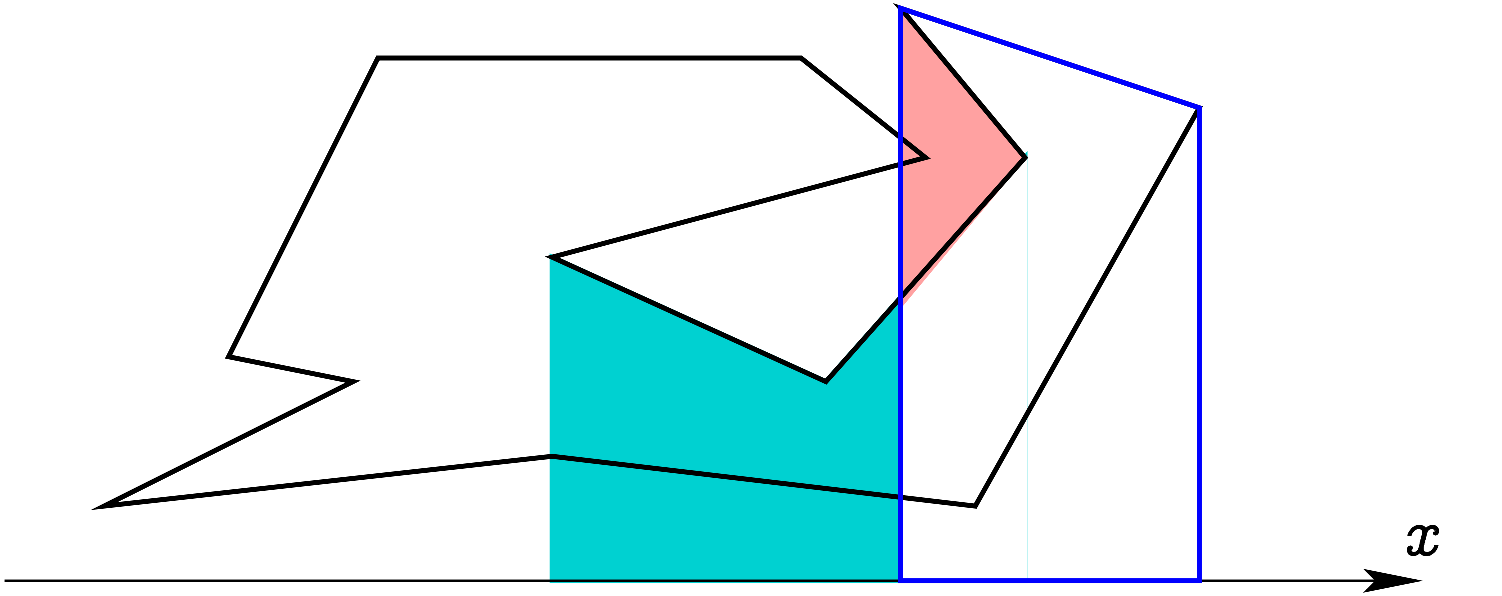

x

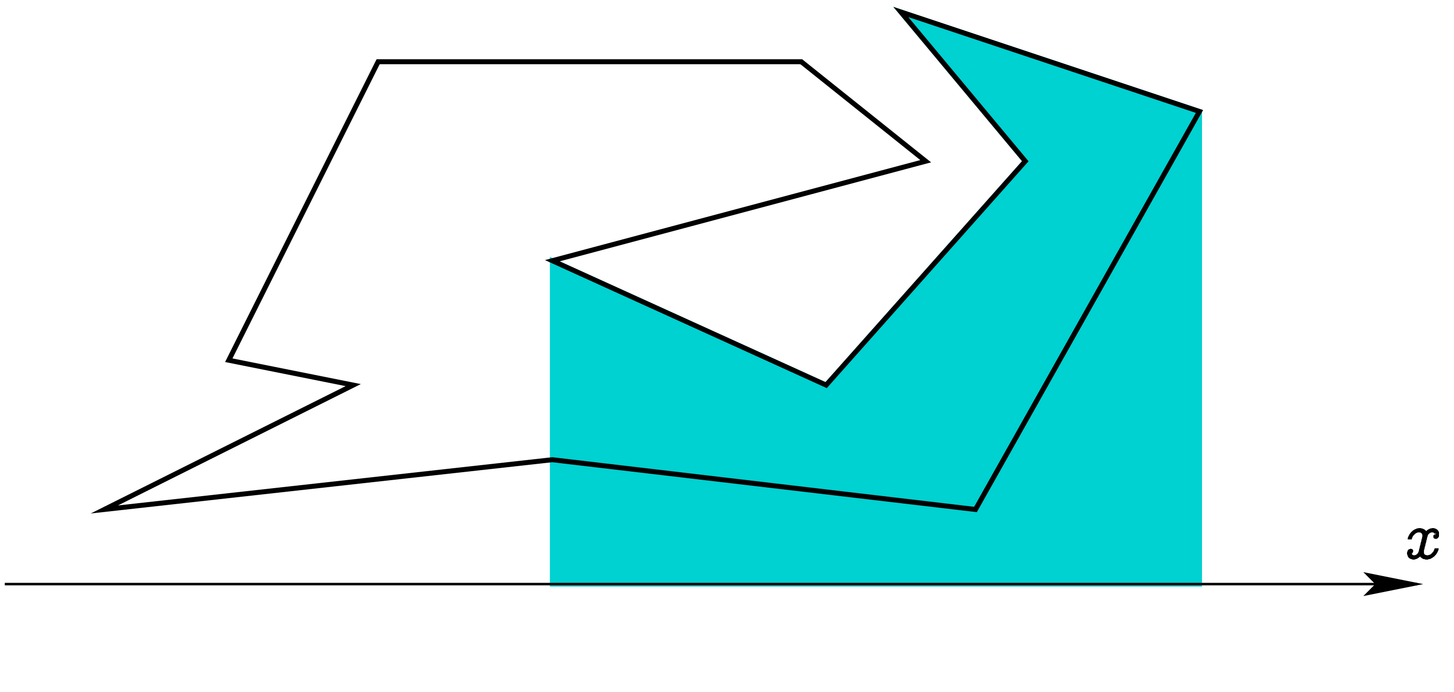

x

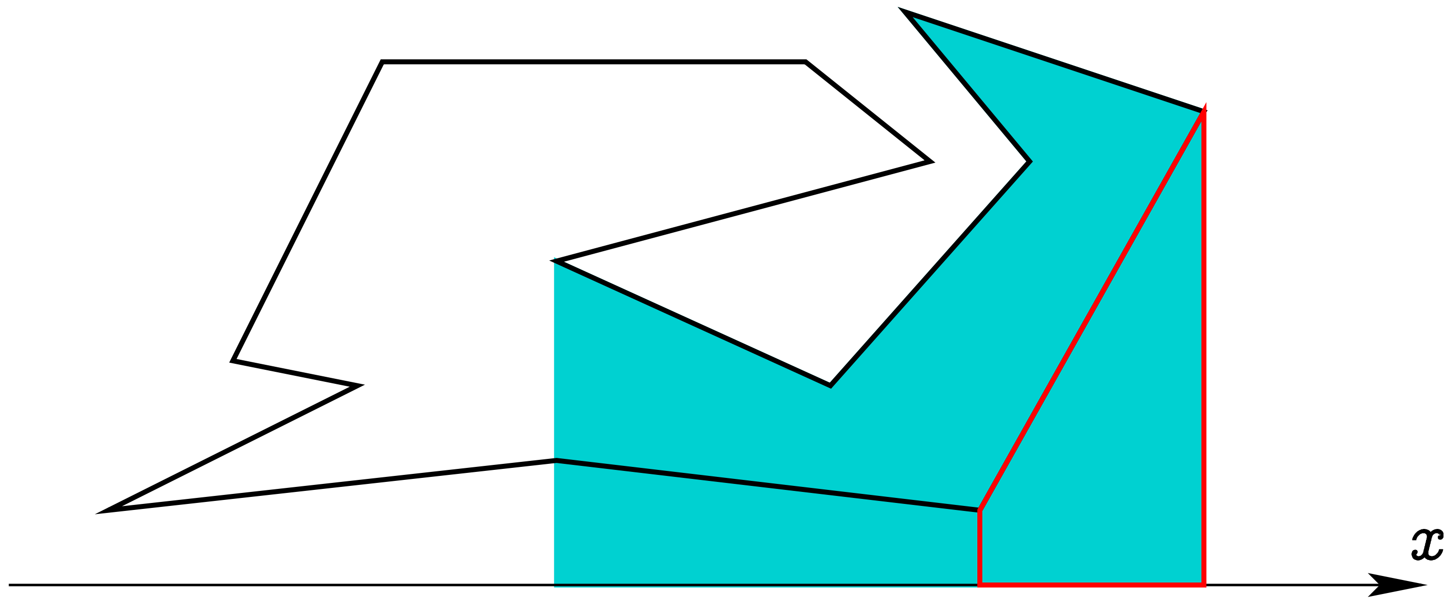

x

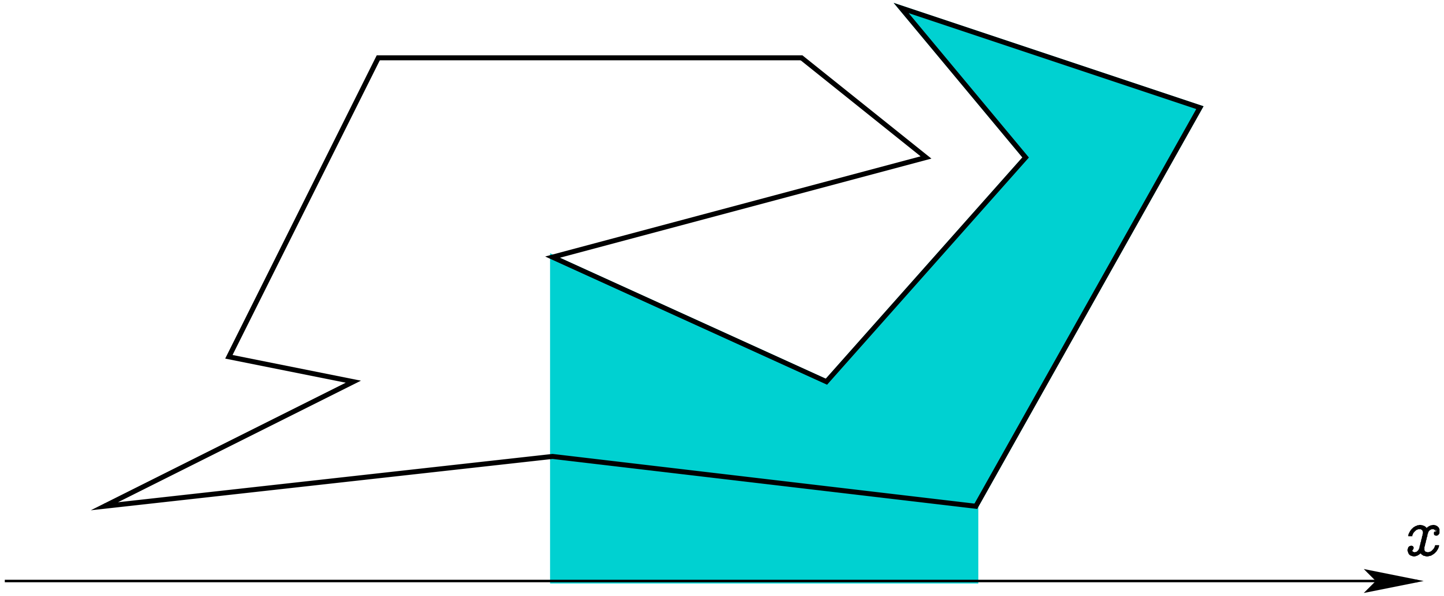

x

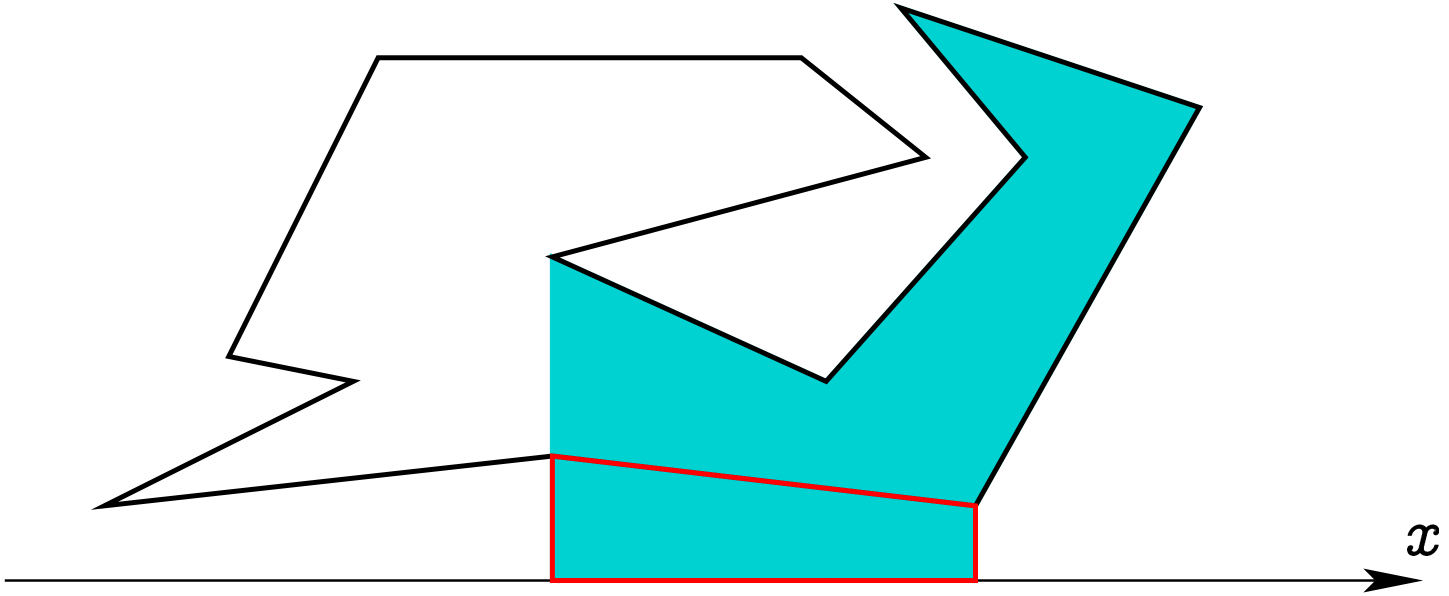

x

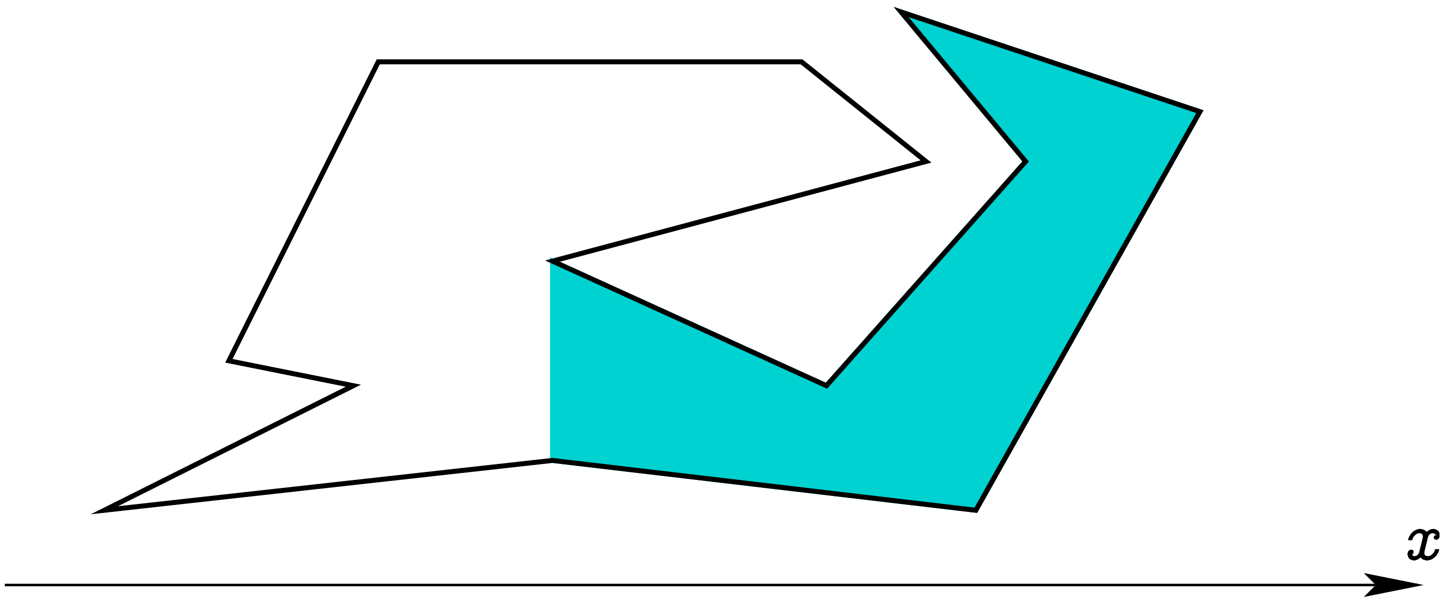

x

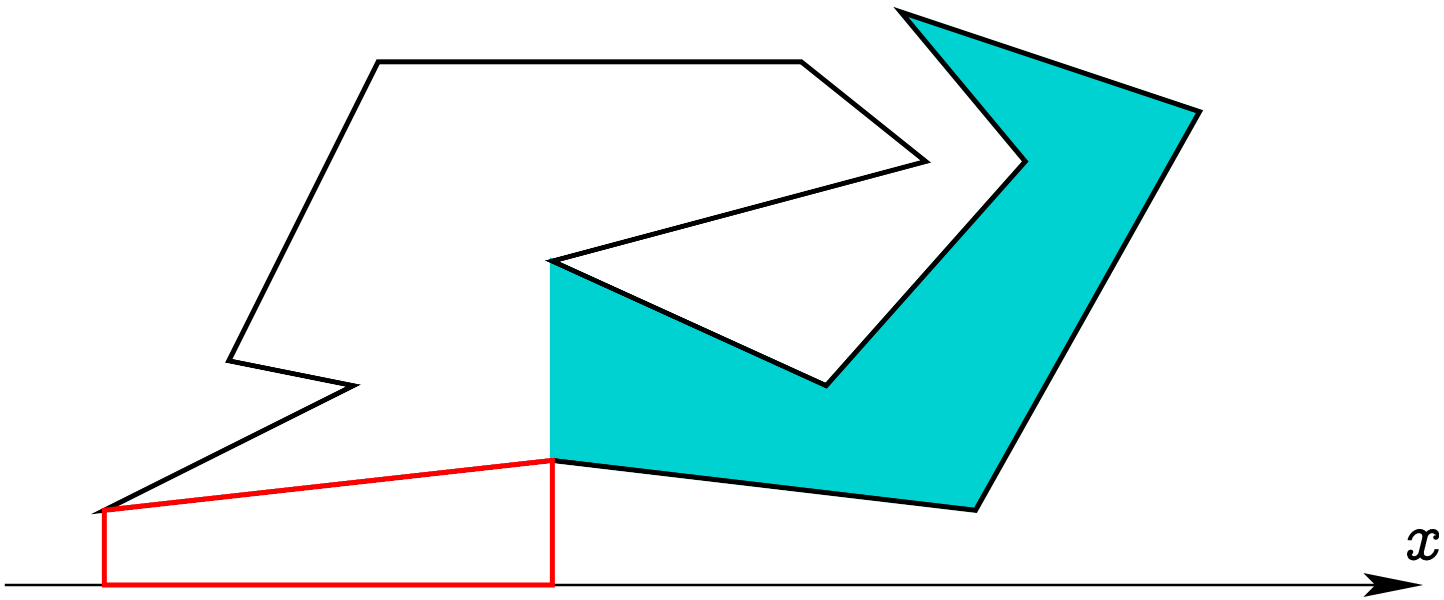

x

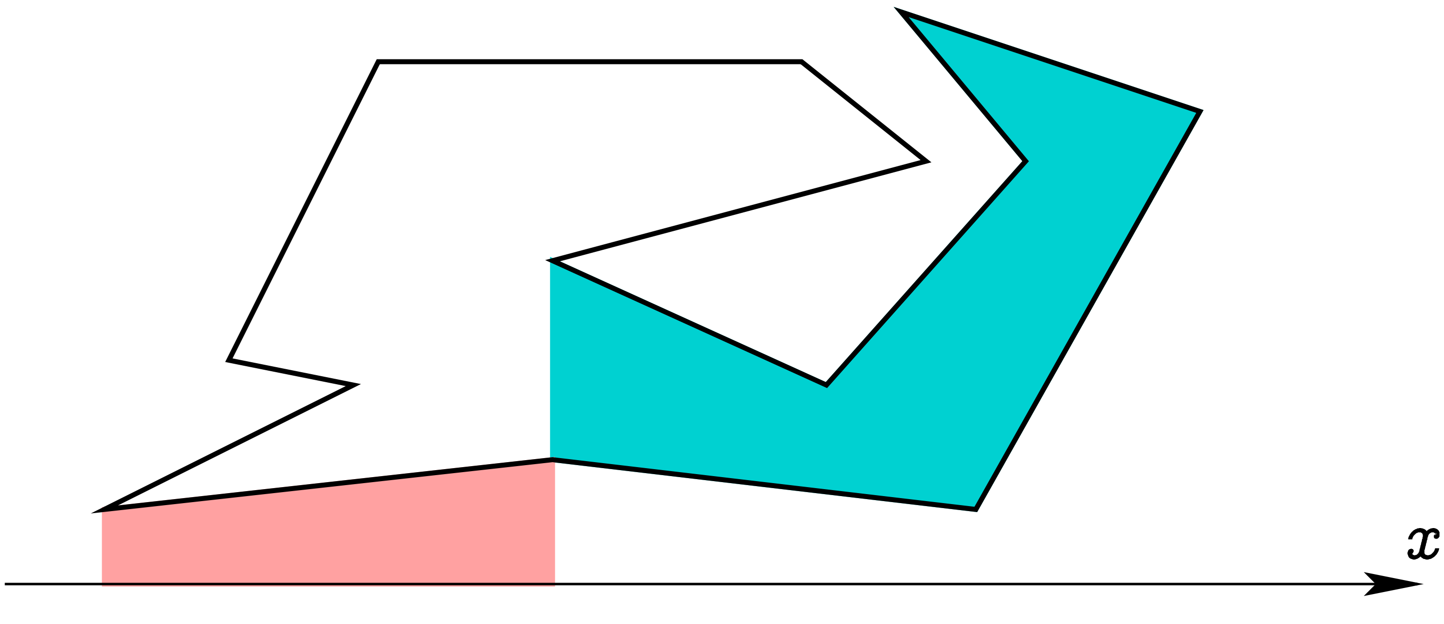

x

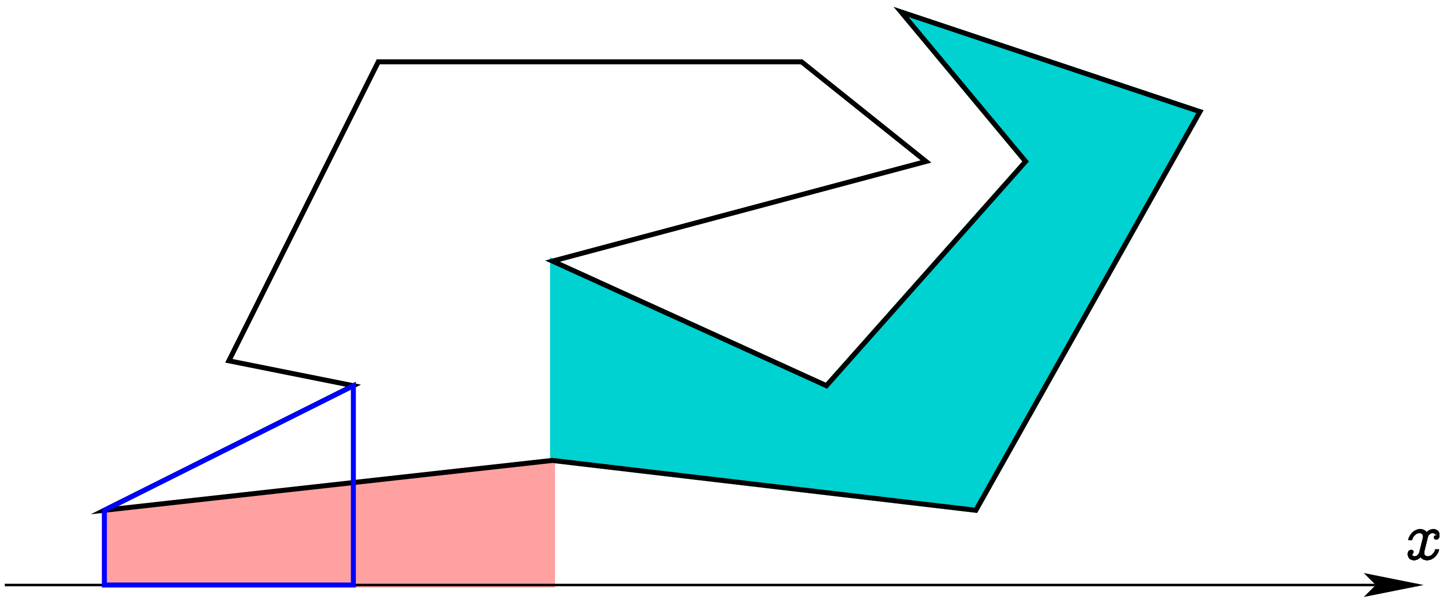

x


x

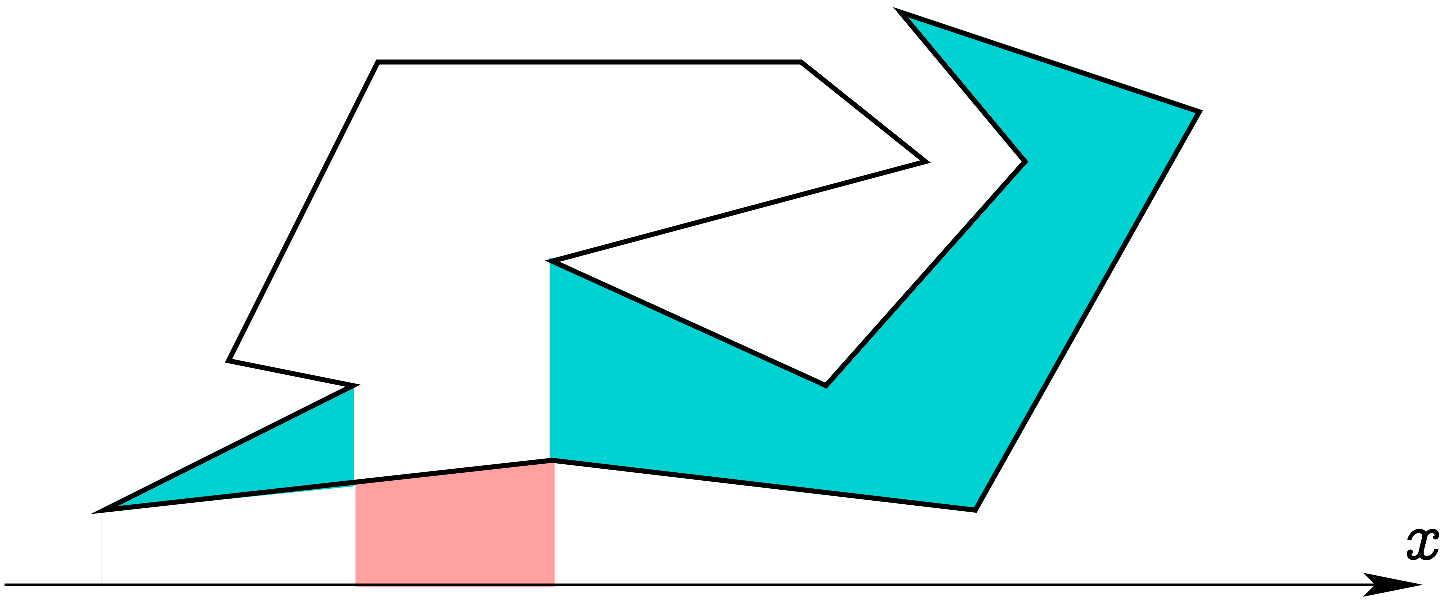

x

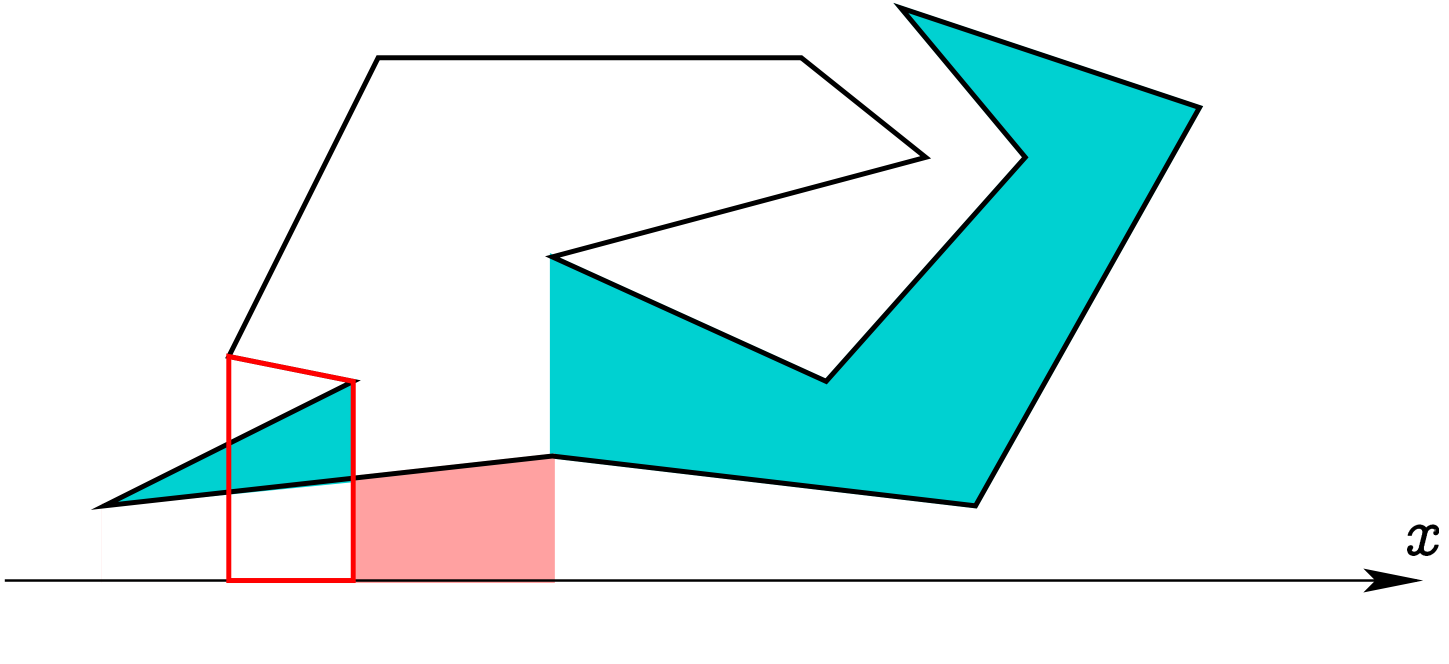

x

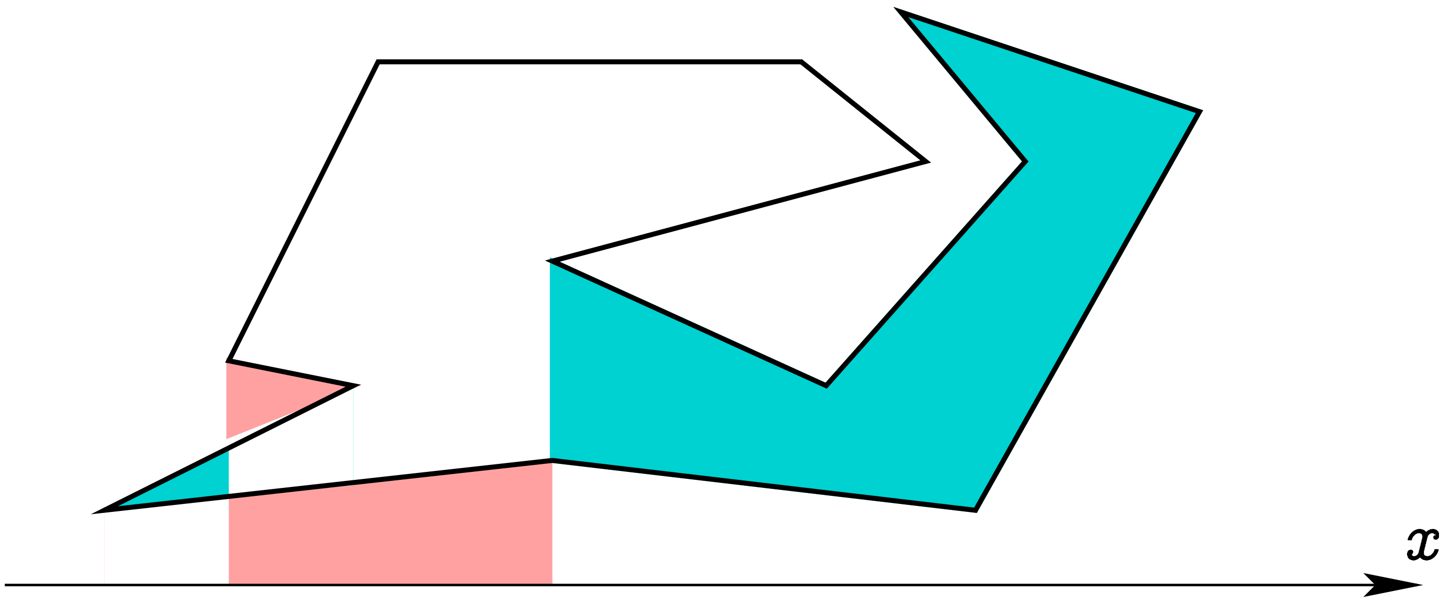

x



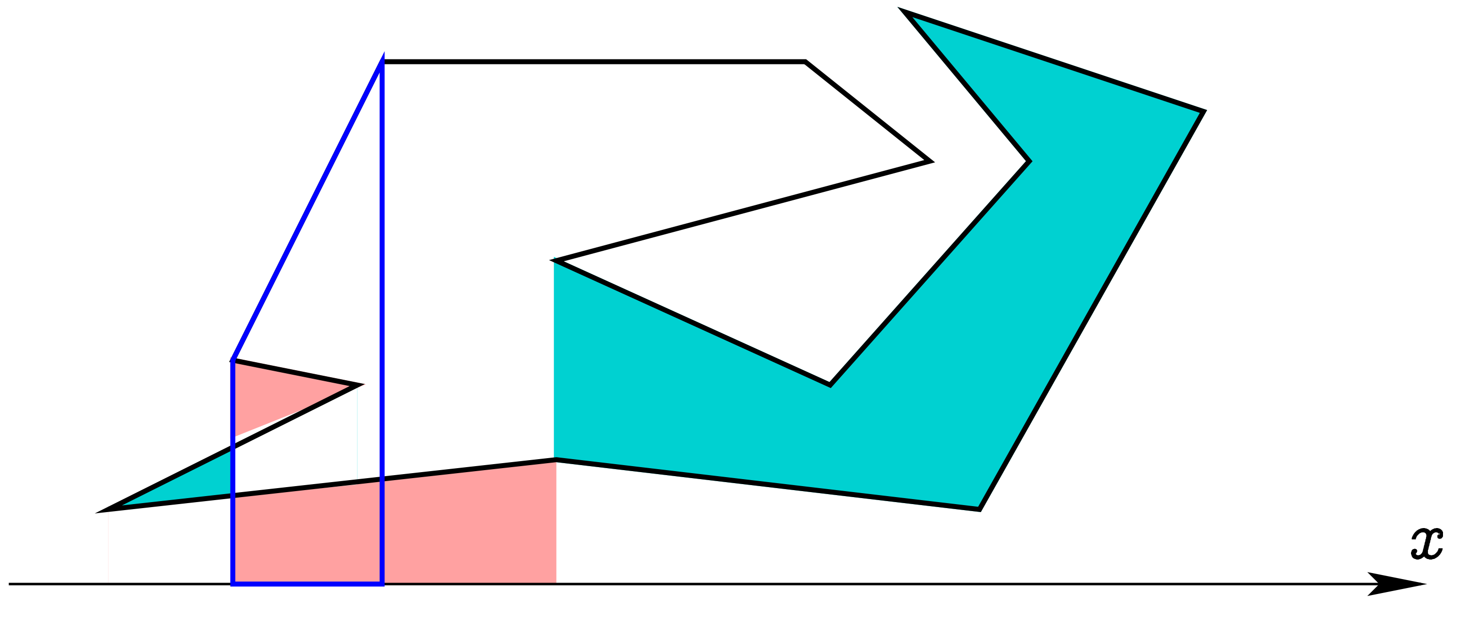
x


x

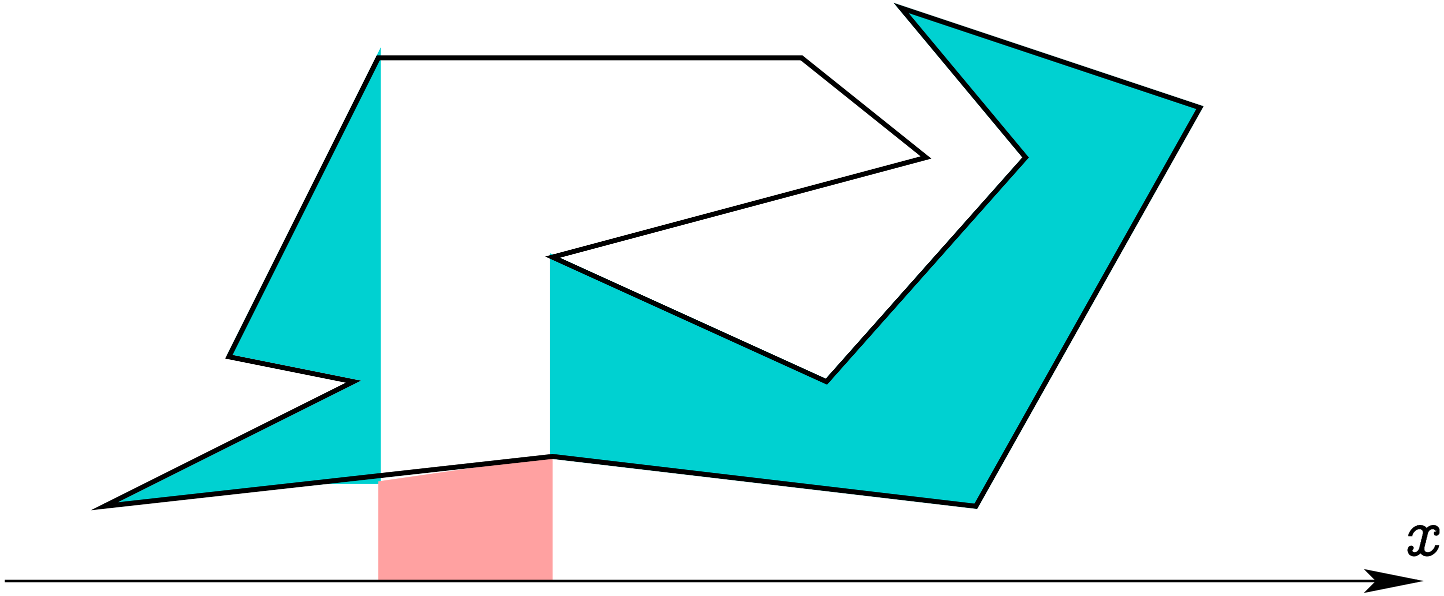

x

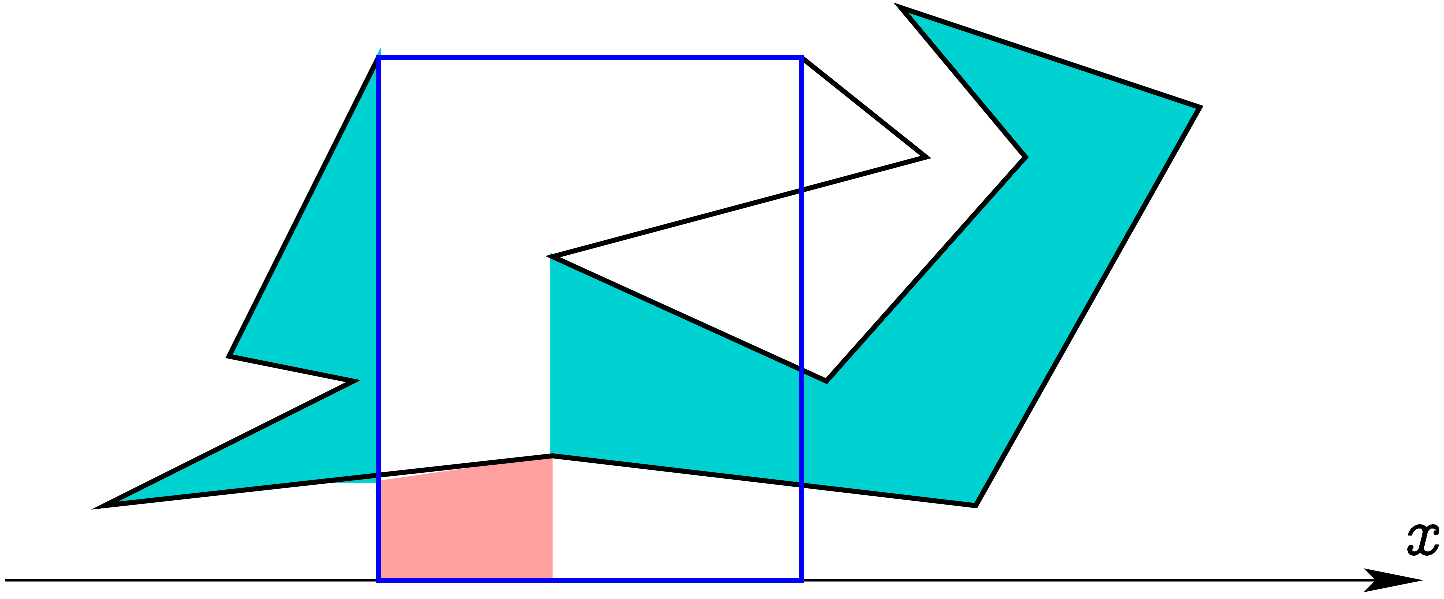


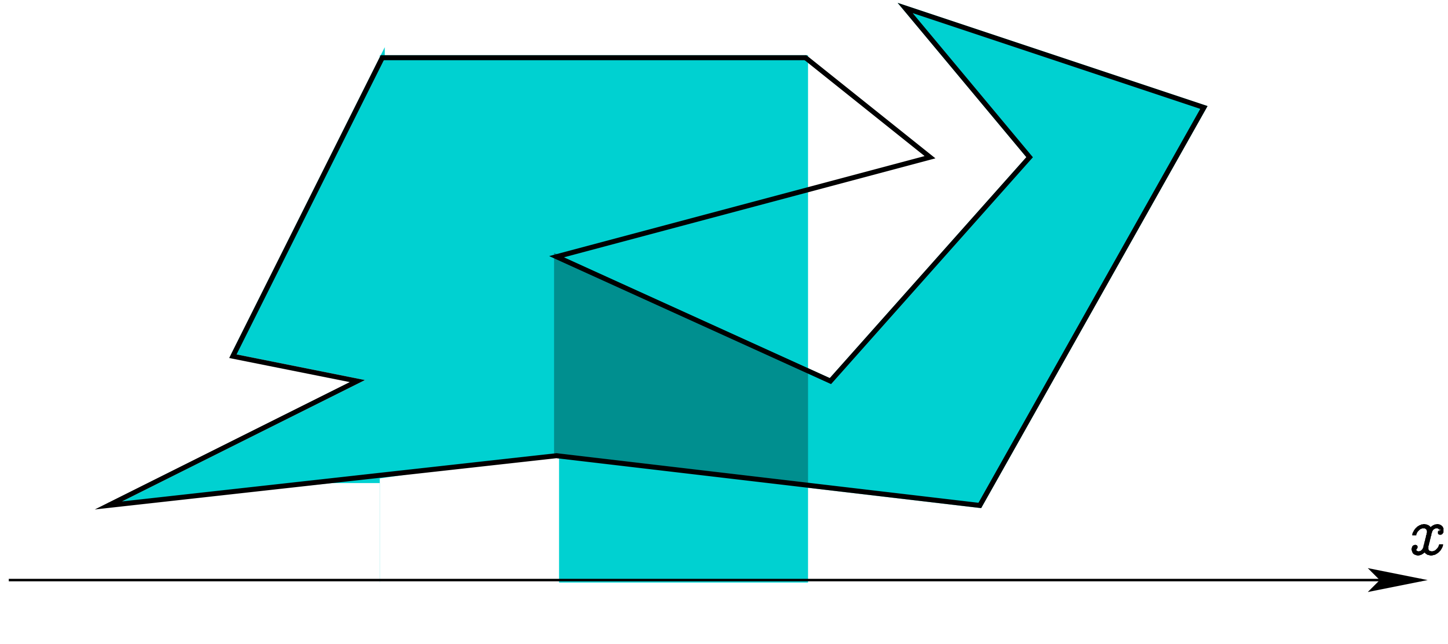
x

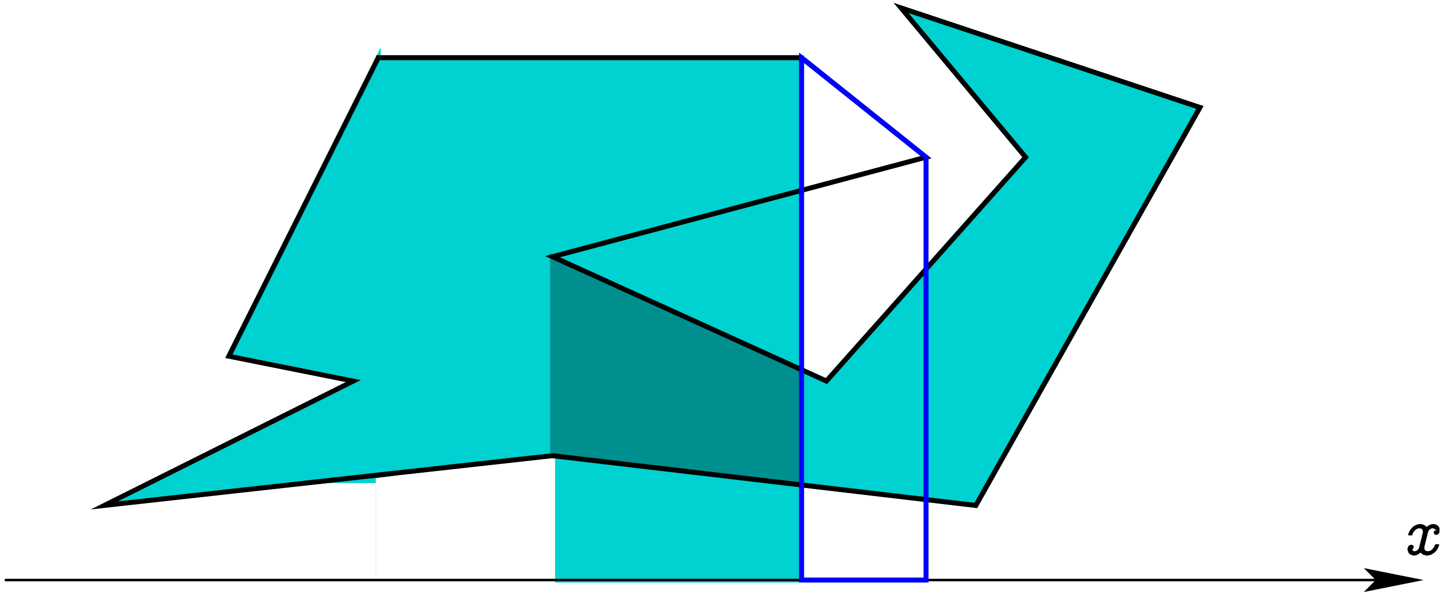


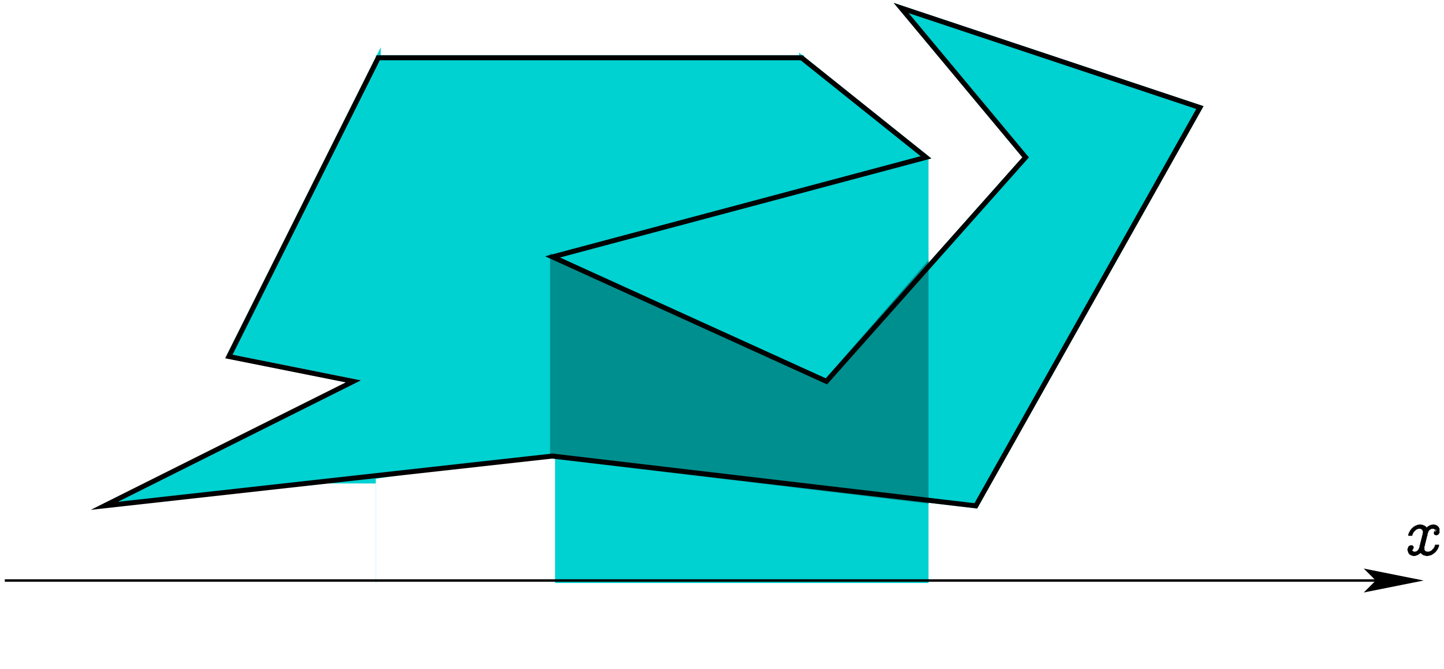
x

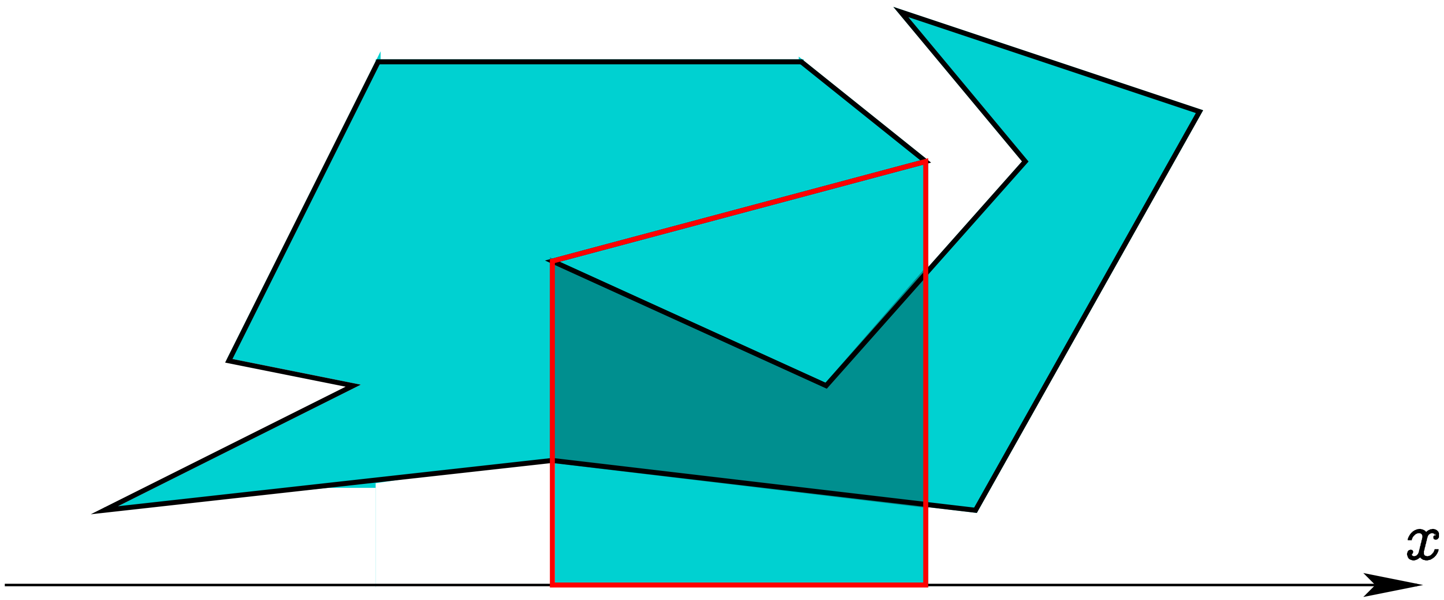

x

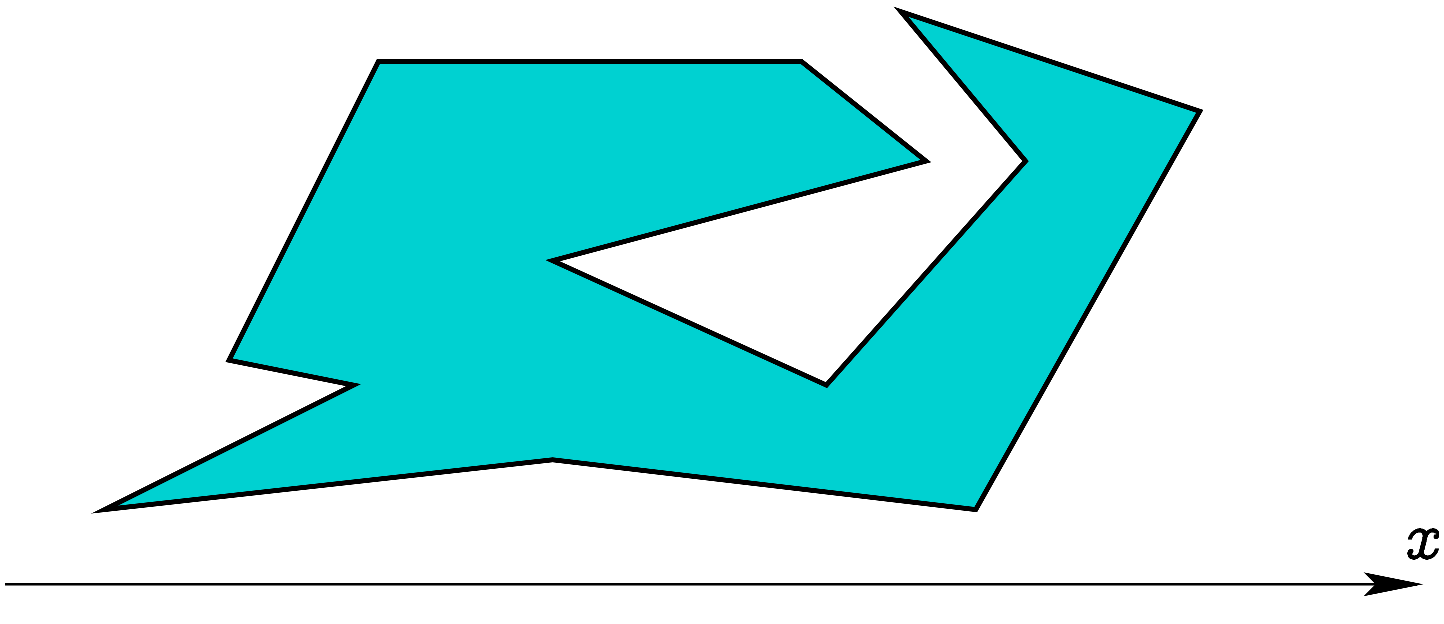

x


x


x


x

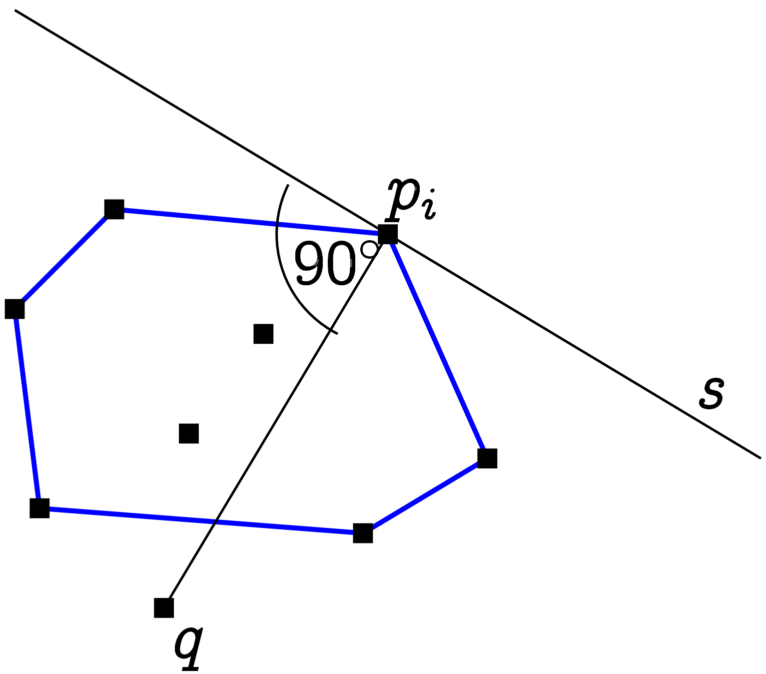

x


x

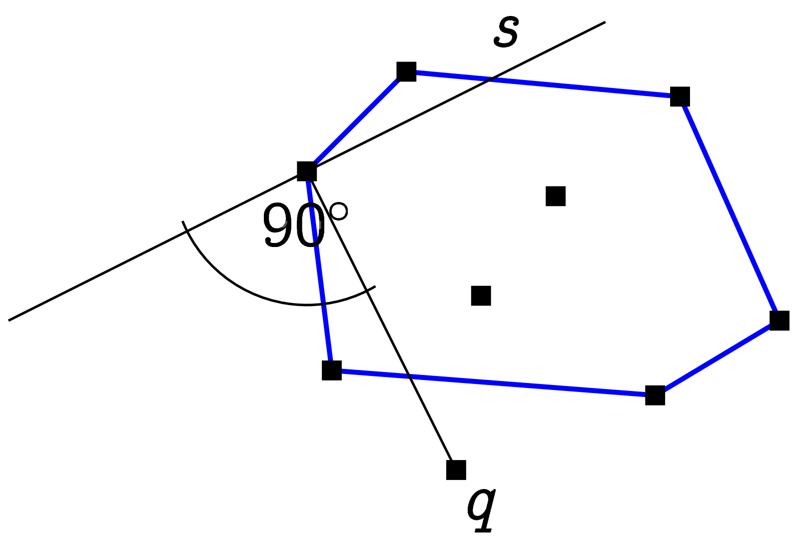
Antud punktid $p_1, \ldots, p_n$. Leida nende seast kaks teineteisest kõige kaugemat punkti.

On üsna ilmne, et need kaks punkti peavad asuma nende punktide kumeral kattel, aga see järeldub ka järgmisest lausest.

Lause. Olgu q mingi punkt ning olgu p_i punktist q kaugeim punkt punktide $p_1, \ldots, p_n$ seas. Olgu s sirge, mis on risti lõiguga $\overline{qp_i}$ ja läbib punkti p_i. Siis s-i ja punktide $p_1, \ldots, p_n$ kumera katte ainus ühine punkt on p_i.

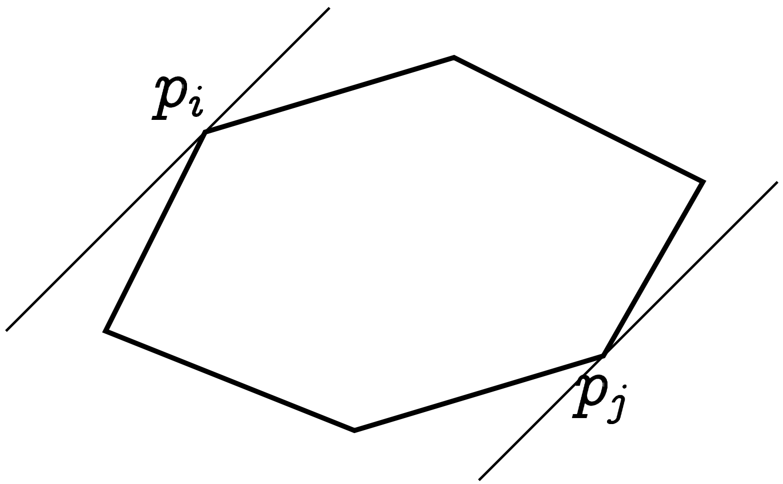


Vastasel juhul võiks mööda katte sisse jäävat s-i osa liikudes q-st kaugemate punktideni jõuda.

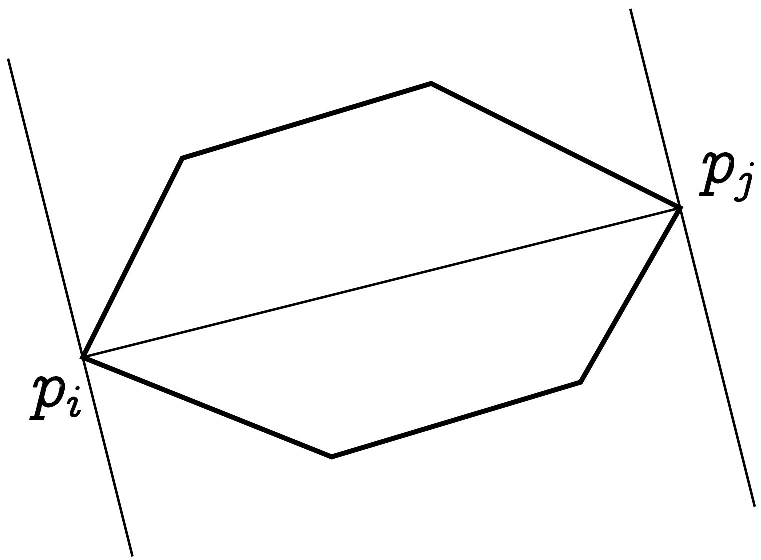


Olgu meil antud kumer hulknurk $p_1 p_2 \cdots p_n$, olgu punktid $p_1, \ldots, p_n$ kellaosuti liikumise vastassuunas. (Grahami seiremeetod annab kumera katte just sellises järjekorras) Loeme, et tal ei ole paralleelseid külgi.

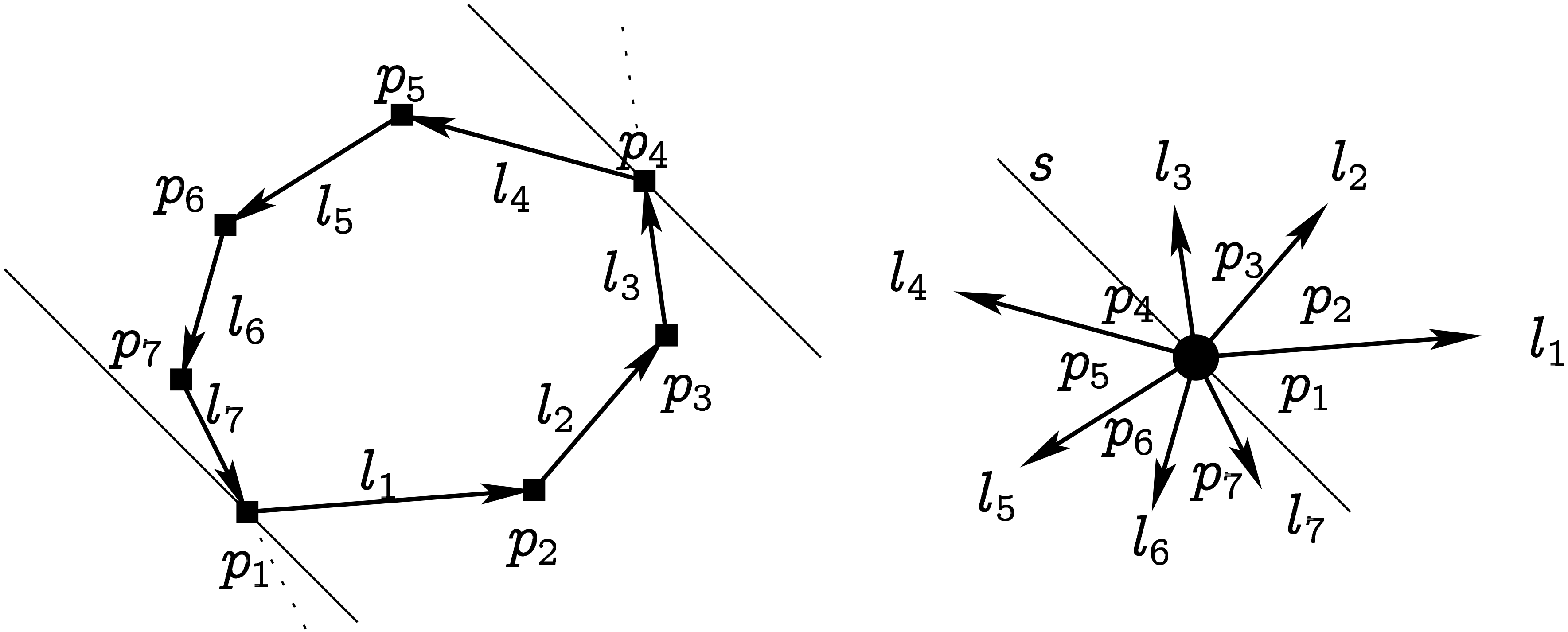
Punktid p_i ja p_j on *antipoodsed*, kui leiduvad kaks paralleelset sirget nii, et ühe ühisosaks selle hulknurgaga on p_i ja teise ühisosaks selle hulknurgaga on p_j.



Lause. Kui p_i ja p_j on teineteisest kõige kaugemal asetsevad punktid, siis nad on antipoodsed.



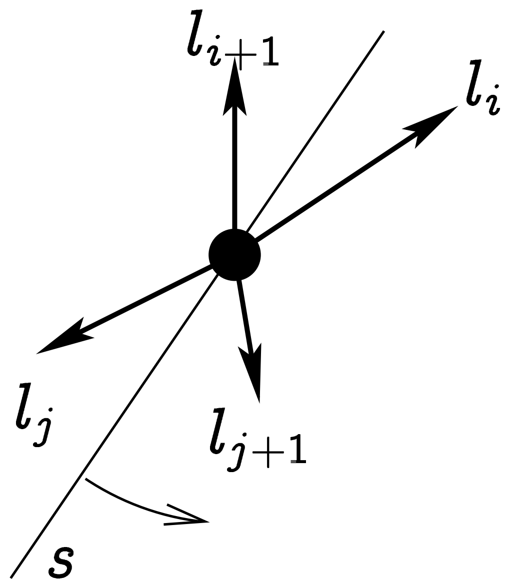
Sobivad näiteks punktides p_i ja p_j võetud lõiguga $\overline{p_i p_j}$ ristuvad sirged.



Antipoodsete paaride leidmine.

Keerutame sirget s.

Kui s on ühelt poolt l_i ja l_{i+1} vahel ja teiselt poolt l_j ja l_{j+1} vahel, siis



huvitab meid, kumba pidi tuleb keerata l_{i+1}-e, et viia ta samasuunaliseks l_{j+1}-ga.

Selle järgi otsustame, kas järgmine piirkond on l_{i+1} ja l_{i+2} vahel või l_{j+1} ja l_{j+2} vahel.

Olgu meil antud objektid $x_1, \ldots, x_n$. Me soovime pidada andmestruktuuri, mis vastaks hulkade komplektile $(S_1, \ldots, S_k)$, nii et iga element x_i kuuluks täpselt ühte hulkadest S_j.

Me soovime järgmiseid operatsioone:

- tee_klass(x) lisab juba vaadeldavatele objektidele uue objekti x, samuti lisab ta uue hulka S ja võtab x-i selle ainsaks elemendiks.
- ühenda(x, y) ühendab hulgad, millesse kuuluvad x ja y, üheks hulgaks.
- leia_esindaja(x) tagastab selle hulga S, kuhu x kuulub, mingi elemendi. Seejuures tagastab ta hulga S iga elemendi jaoks sama elemendi.

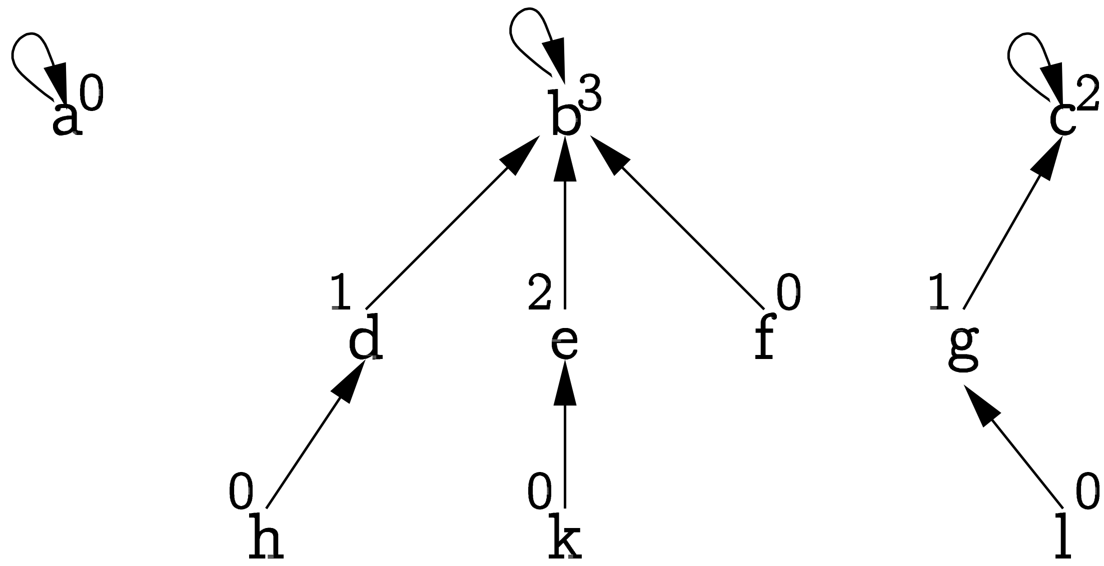
*Galler-Fisher*i meetodil klasside kujutamisel on igal objektil kaks abivälja — $.viit$ ja $.h$.

Klassi kujutatakse puuna tema elementidest, $.viit$ on viit ülemusele. Objekti väli $.h$ on mingi naturaalarv, mis on vähemalt sama suur kui alampuu, mille juureks see objekt on, kõrgus.

Kui mingi objekt on klassi kujutava puu juureks, siis tema $.viit$ viitab talle enesele.

leia_esindaja(x) tagastab x-i sisaldava puu juure.

Näiteks võib ({a}, {b,d,e,f,h,k}, {c,g,l}) kujutatud olla järgmisel viisil:



tee_klass(x) on

$x.viit := x$; $x.h := 0$

ühenda(x, y) on

lingi(leia_esindaja(x), leia_esindaja(y))

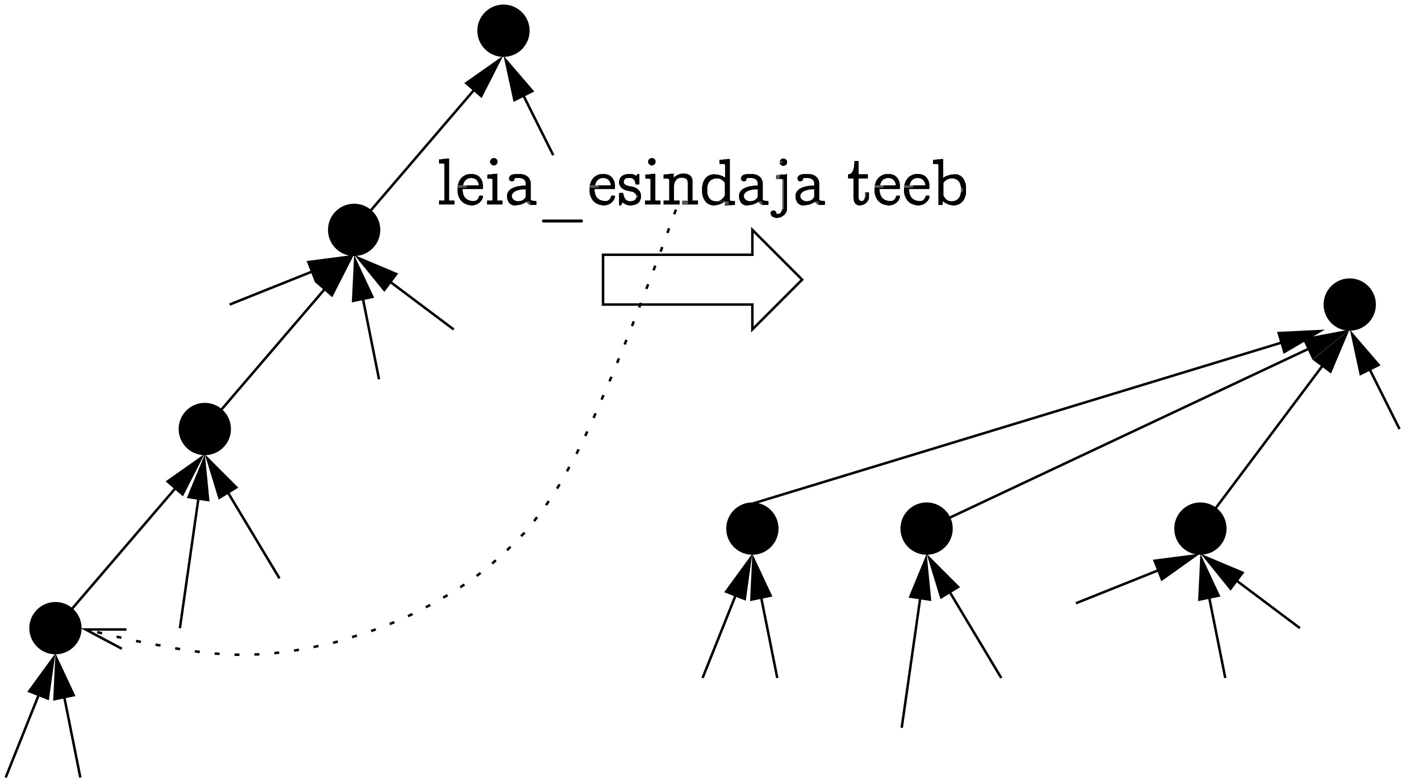
lingi(x, y) on

if $x.h > y.h$ **then** $y.viit := x$ **else** $x.viit := y$

if $x.h = y.h$ **then** $y.h := y.h + 1$

leia_esindaja(x) on

```
if x ≠ x.viit then
    x.viit := leia_esindaja(x.viit)
return x.viit
```



Keerukus: kui meil on n objekti ja m operatsiooni tee_klass, ühenda ja leia_esindaja, siis ajakulu on $O(m \log^* n)$.

$\log^* n$ on selline i, et $0 < \underbrace{\log \cdots \log}_{i} n \leqslant 1$.

n-i vahemik	$\log^* n$
$n = 1$	0
$n = 2$	1
$3 \leqslant n \leqslant 4$	2
$5 \leqslant n \leqslant 16$	3
$17 \leqslant n \leqslant 65536$	4
$65537 \leqslant n \leqslant 2^{65536}$	5

Me näitame, et n objekti korral on m operatsiooni tee_klass, lingi ja leia_esindaja ajakulu on $O(m \log^* n)$.

Siis on ka m operatsiooni tee_klass, ühenda ja leia_esindaja ajakulu $O(m \log^* n)$, sest nad sisaldavad ülimalt $3m$ operatsiooni tee_klass, lingi ja leia_esindaja.

Puude struktuuri ja selle muutumise kohta töö jooksul võib öelda järgmist. Olgu x mingi objekt.

- Alguses on x mingi puu juur. Kui ta töö käigus mingil hetkel mittejuureks muutub, siis ei saa ta enam kunagi juureks.
- Kirjutusviis $x.viit = x$ tähendab „x on mingi puu juur".

Mõned tulemused väljade $.h$ väärtuste kohta ja nende muutumise kohta töö jooksul. Vaatame mingit objekti x.

- Kui $x.viit \neq x$, siis $x.h < x.viit.h$.
- Alguses on $x.h = 0$. Töö jooksul ta ei vähene. Alates hetkest, kus $x.viit \neq x$, $x.h$ enam ei muutu.
- $x.viit.h$ töö jooksul ei vähene.
- Olgu $P(x)$ tippude arv alampuus, mille juureks on x. Siis $P(x) \geqslant 2^{x.h}$.

Tõestused on induktsiooniga üle tehtud operatsioonide arvu.

Kehtigu eelmisel slaidil toodud väited peale mingit arvu operatsioone. Teeme järjekordse operatsiooni.

Kui see oli tee_klass(x), siis saab $x.h$ väärtuseks 0. x on mingi puu juur. $P(x) = 1 = 2^{x.h}$.

Kui see oli lingi(x, y), siis muutused toimuvad ainult x-i ja y-i juures.

Kui enne operatsiooni $x.h \neq y.h$ (üldisust kitsendamata loeme, et $x.h < y.h$), siis pärast operatsiooni on juureks mittevõetud tipu x väli $.h$ väiksem kui juureks võetud tipu y väli $.h$.

Samuti on pärast operatsiooni $P(y) \geqslant 2^{y.h} + 2^{x.h} \geqslant 2^{y.h}$.

Kui enne operatsiooni $x.h = y.h$, siis pärast $y.h = x.h + 1 > x.h$. Samuti $P(y) \geqslant 2^{y.h-1} + 2^{x.h} = 2 \cdot 2^{y.h-1} = 2^{y.h}$.

lingi on ainus operatsioon, mis muudab välja $.h$ väärtusi. Muudetakse tipul, mis on puu juur.

Kui järjekordne operatsioon oli leia_esindaja(x), siis võis $x.viit$ hakata viitama kõrgemale kui varem.

Kuna ennem oli $x.h < x.viit.h$ iga tipu x jaoks, siis puus ülespoole liikudes välja $.h$ väärtused suurenevad.

Välja $x.viit$ muutes siis $x.viit.h$ suurenes.

Järeldus. Tippe, mille välja $.h$ väärtus on vähemalt r, on ülimalt $n/2^r$.

Tõestus. Kui mingi tipu x jaoks omistatakse $x.h := r$, siis peab puus, mille juureks on x, olema vähemalt 2^r tippu. Nende tippude (v.a. x ise) aste on väiksem kui r ja enam ei muutu.

Kui mõne teise tipu y jaoks omistatakse $y.h := r$, siis peab puus, mille juures on y, olema vähemalt 2^r tippu. Need tipud peavad olema erinevad puu, mille juureks on x, tippudest, sest x ei kuulu puusse, mille juureks on y.

Seega saab ülimalt iga 2^r-s tipp saada $.h$ väärtuseks r või enam. Muuhulgas on siis iga tipu välja $.h$ väärtus $\leqslant \log n$.

Üks operatsioon tee_klass või lingi vajab konstantset aega.

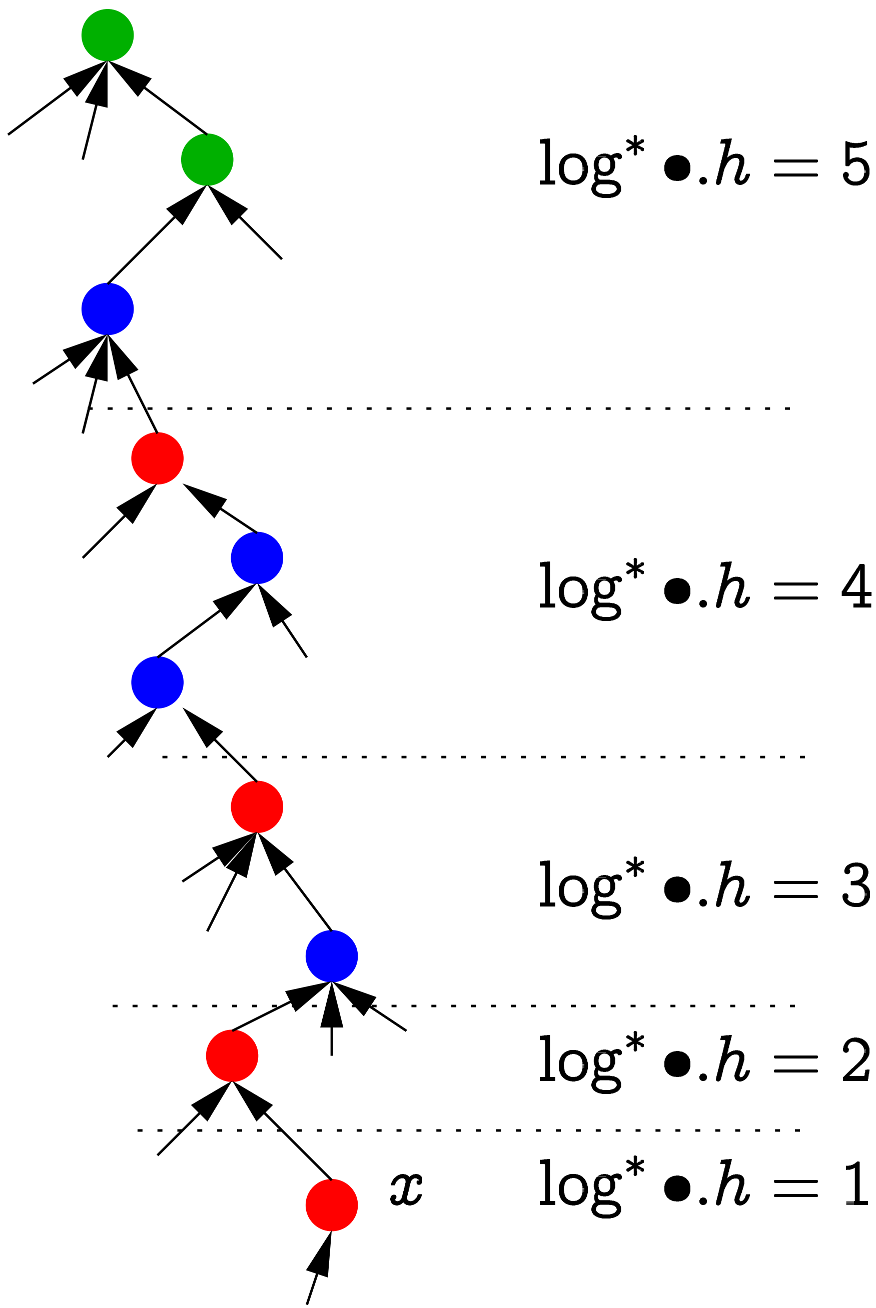
Ühe operatsiooni leia_esindaja(x) tööaeg on proportsionaalne objekti x kaugusega teda sisaldava puu juurest selle operatsiooni tegemise hetkel.

Me näitame, et peale m-i operatsiooni on nende kauguste summa $O(m \log^* n)$.

Vaatame mingit operatsiooni leia_esindaja(x).

Vaatame ahelat tipust x teda sisaldava puu juureni. Värvime sellel ahelal tipud järgmiselt:

- Juure ja tema vahetu järglase värvime roheliseks.
- Mõne teise tipu v sellelt ahelalt värvime punaseks parajasti siis, kui $\log^* y.h \neq \log^* y.viit.h$.
  - Defineerime $\log^* 0 := -1$.
- Ülejäänud tipud värvime siniseks.
  - Neil tippudel $\log^* y.h = \log^* y.viit.h$.



$\log^* \bullet.h = 5$
$\log^* \bullet.h = 4$
$\log^* \bullet.h = 3$
$\log^* \bullet.h = 2$
x
$\log^* \bullet.h = 1$

Igal operatsioonil leia_esindaja on ülimalt kaks rohelist ja ülimalt $\log^* \log n = \log^* n - 1$ punast tippu.

Kui mingi tipp x on kunagi olnud punane, siis ei saa ta enam kunagi edaspidi olla sinine, sest

- $x.h$ enam ei muutu, seega ei muutu ka $\log^* x.h$;
- $\log^* x.h < \log^* x.viit.h$;
- $x.viit.h$ saab üksnes suureneda, seega ka $\log^* x.viit.h$ saab üksnes suureneda.

Uurime, mitu korda saab üks tipp olla sinine.

Kui tipp x on mingis operatsioonis leia_esindaja sinine, siis muutub tema otsene ülemus. S.t. $x.viit.h$ suureneb selle operatsiooni käigus.

Ta saab suureneda ainult senikaua kuni $\log^* x.viit.h = \log^* x.h$, muidu muutub tipp punaseks.

Olgu $S(x)$ kordade arv, mil tipp x võib olla sinine. Siis on m operatsiooni tehes operatsioonidel leia_esindaja vaadeldavate ahelate pikkuste summa ülimalt

$$2m + m(\log^* n - 1) + \sum_{i=1}^{n} S(x_i) \quad .$$

Tähistame: $\mathbf{2}^{*-2} := -1$, $\mathbf{2}^{*-1} := 0$, $\mathbf{2}^{*0} := 1$, $\mathbf{2}^{*i} := 2^{\mathbf{2}^{*i-1}}$.

S.t. $\mathbf{2}^{*i} = \left. 2^{2^{\cdot^{\cdot^{\cdot^{2}}}}} \right\} i$

Siis $\log^* n = i$ parasjagu siis, kui $\mathbf{2}^{*i-1} + 1 \leqslant n \leqslant \mathbf{2}^{*i}$.

Olgu $j = \log^* x.h$. Siis $x.viit.h$ võib suureneda ülimalt $\mathbf{2}^{*j} - \mathbf{2}^{*j-1} - 1$ korral, nii et $\log^* x.viit.h$ ei saa suuremaks kui $\log^* x.h$.

S.t. $S(x) \leqslant \mathbf{2}^{*\log^* x.h} - \mathbf{2}^{*(\log^* x.h)-1} - 1$.

Olgu $N(j) = |\{x : \log^* x.h = j\}|$. Siis on m operatsiooni tehes operatsioonidel leia_esindaja vaadeldavate ahelate pikkuste summa ülimalt

$$2m + m(\log^* n - 1) + \sum_{j=-1}^{\log^* n - 1} N(j)(\mathbf{2}^{*j} - \mathbf{2}^{*j-1} - 1) .$$

Peale selle,

$$N(j) \leqslant \sum_{r=\mathbf{2}^{*j-1}+1}^{\mathbf{2}^{*j}} \frac{n}{2^r},$$

sest tippe, mille välja $.h$ väärtus on vähemalt r, on ülimalt $n/2^r$.

$$
\begin{aligned}
N(j) &\leqslant \sum_{r=\mathbf{2}^{*j-1}+1}^{\mathbf{2}^{*j}} \frac{n}{2^r} = \frac{n}{2^{\mathbf{2}^{*j-1}+1}} \sum_{r=0}^{\mathbf{2}^{*j}-\mathbf{2}^{*j-1}-1} \frac{1}{2^r} \\
&\leqslant \frac{n}{2^{\mathbf{2}^{*j-1}+1}} \sum_{r=0}^{\infty} \frac{1}{2^r} = \frac{n \cdot 2}{2^{\mathbf{2}^{*j-1}+1}} = \frac{n}{2^{\mathbf{2}^{*j-1}}} \\
&= \begin{cases} 2n, & \text{kui } j = -1 \\ \dfrac{n}{\mathbf{2}^{*j}}, & \text{kui } j \geqslant 0 \end{cases}
\end{aligned}
$$

$$2m + m(\log^* n - 1) + \sum_{j=-1}^{\log^* n - 1} N(j)(\mathbf{2}^{*j} - \mathbf{2}^{*j-1} - 1) \leqslant$$

$$m(\log^* n + 1) + \sum_{j=-1}^{\log^* n - 1} N(j)\mathbf{2}^{*j} =$$

$$m(\log^* n + 1) + 2n\mathbf{2}^{*-1} + \sum_{j=0}^{\log^* n - 1} \frac{n}{\mathbf{2}^{*j}}\mathbf{2}^{*j} =$$

$$m(\log^* n + 1) + \sum_{j=0}^{\log^* n - 1} n =$$

$$m(\log^* n + 1) + n \log^* n = O(m \log^* n)$$