Rúmfræði

Bergur Snorrason

4. apríl 2019

Efnisyfirlit

Google Code Jam

- Aðfaranótt 6. apríl hefst fyrsta umferð í Google Code Jam.

Google Code Jam

- Aðfaranótt 6. apríl hefst fyrsta umferð í Google Code Jam.
- Hún stendur yfir í 24 tíma.

Google Code Jam

- Aðfaranótt 6. apríl hefst fyrsta umferð í Google Code Jam.
- Hún stendur yfir í 24 tíma.
- Ég mæli með.

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*
  - *Planetaris*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*
  - *Planetaris*
  - *Strikercount*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*
  - *Planetaris*
  - *Strikercount*
  - *Tildes*

Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*
  - *Planetaris*
  - *Strikercount*
  - *Tildes*
- Keppnin verður með svipuðu sniði og síðast.

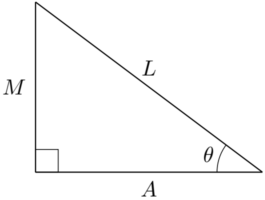
Lokakeppnin

- Eftir viku verður lokakeppnin okkar.
- Dæmin verða sex talsnins:
  - *Assosiation of Myths*
  - *Digbuild*
  - *Geezer Scripts*
  - *Planetaris*
  - *Strikercount*
  - *Tildes*
- Keppnin verður með svipuðu sniði og síðast.
- Til að fá skil þarf að leysa eitt dæmi.

Efnisyfirlit

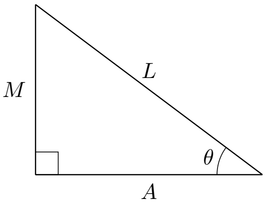
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.



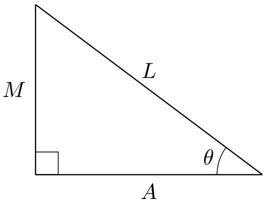
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.



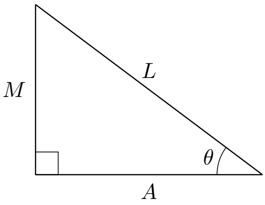
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.
- Fyrir rétthyrnda þríhyrninga gildir:



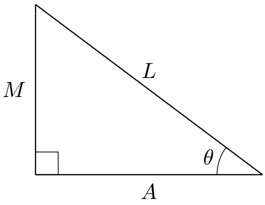
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.
- Fyrir rétthyrnda þríhyrninga gildir:
  - $\dfrac{A}{L} = \cos\theta$.



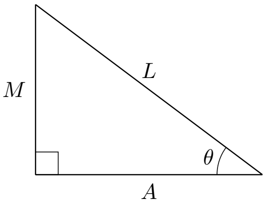
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.
- Fyrir rétthyrnda þríhyrninga gildir:
  - $\dfrac{A}{L} = \cos\theta$.
  - $\dfrac{M}{L} = \sin\theta$.



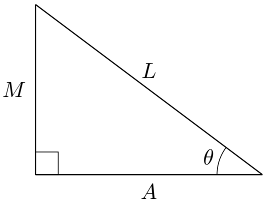
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.
- Fyrir rétthyrnda þríhyrninga gildir:
  - $\dfrac{A}{L} = \cos\theta$.
  - $\dfrac{M}{L} = \sin\theta$.
  - $\dfrac{M}{A} = \dfrac{\sin\theta}{\cos\theta} = \tan\theta$.



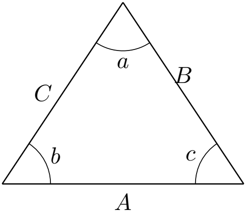
Hornaföll

- Þessi glæra ætti að vera upprifjun fyrir flest ykkar.
- Þríhyrningur er sagður rétthyrndur ef eitt horna hans er 90°.
- Fyrir rétthyrnda þríhyrninga gildir:
    - $\dfrac{A}{L} = \cos\theta.$
    - $\dfrac{M}{L} = \sin\theta.$
    - $\dfrac{M}{A} = \dfrac{\sin\theta}{\cos\theta} = \tan\theta.$
- Einnig gildir regla Pýthagorasar, $L^2 = A^2 + M^2$.



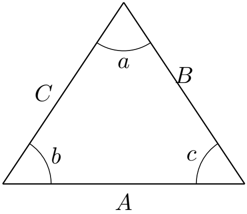
Hornaföll

- Almennar gildir um þríhyrninga:



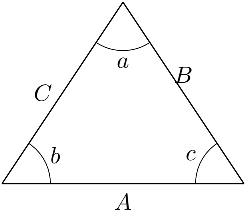
Hornaföll

- Almennar gildir um þríhyrninga:
  - $\dfrac{\sin a}{A} = \dfrac{\sin b}{B} = \dfrac{\sin c}{C}$ (sínus reglan).

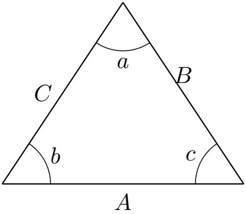


Hornaföll

- Almennar gildir um þríhyrninga:
  - $\dfrac{\sin a}{A} = \dfrac{\sin b}{B} = \dfrac{\sin c}{C}$ (sínus reglan).
  - $A^2 = B^2 + C^2 - 2BC\cos a$ (kósínus reglan)



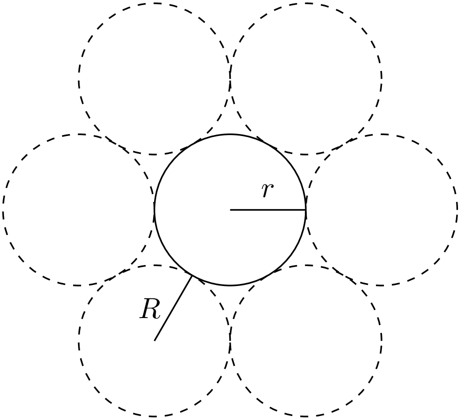
- Almennar gildir um þríhyrninga:
  - $\dfrac{\sin a}{A} = \dfrac{\sin b}{B} = \dfrac{\sin c}{C}$ (sínus reglan).
  - $A^2 = B^2 + C^2 - 2BC\cos a$ (kósínus reglan)
- **Æfing:** Sannið reglu Pýthagorasar með kósínus reglunni.



Sýnidæmi: NN and the Optical Illusion - Codeforces

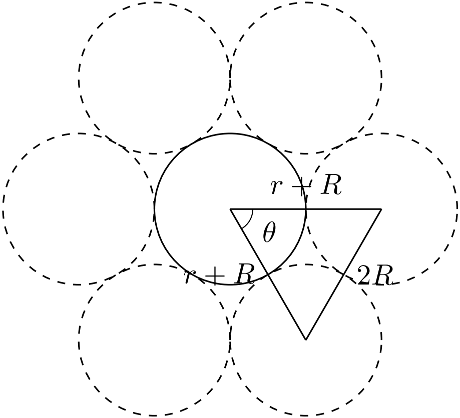
Þér er gefið heiltölu n og rauntölu r. Þú teiknar hring á blað með geilsa r. Þú vilt teikna n jafn stóra hringi í kringum hringinn þinn þannig að þeir skeri hringinn þinn og aðlæga hringi í nákvæmlega einum punkti. Hver þarf geilsi ytri hringjanna að vera.
https://codeforces.com/problemset/problem/1100/C

Ef $n = 6$ fæst eftirfarandi mynd, þar sem R er svarið.



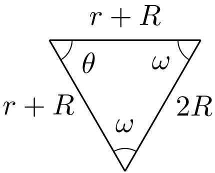
Sýnidæmi: NN and the Optical Illusion - Codeforces

Sjáum að fjarlægðin frá miðjum myndarinnar að miðju ytri hringjanna er $r + R$. Við fáum því eftirfarandi jafnarma þríhyrning.



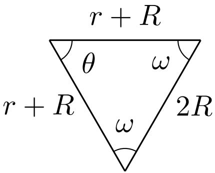
Sýnidæmi: NN and the Optical Illusion - Codeforces

- Hornið θ af síðustu glæru er það eina (ásamt R, að sjálfsögðu) háð n á myndinn, svo almennt þurfum við að finna R út frá eftirfarandi mynd.



Sýnidæmi: NN and the Optical Illusion - Codeforces

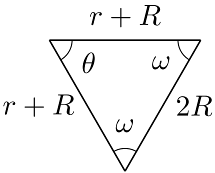
- Hornið θ af síðustu glæru er það eina (ásamt R, að sjálfsögðu) háð n á myndinn, svo almennt þurfum við að finna R út frá eftirfarandi mynd.
- Nú er $\theta = \dfrac{360^\circ}{n}$ og $\omega = \dfrac{180^\circ - \theta}{2}$.



Sýnidæmi: NN and the Optical Illusion - Codeforces

- Hornið θ af síðustu glæru er það eina (ásamt R, að sjálfsögðu) háð n á myndinn, svo almennt þurfum við að finna R út frá eftirfarandi mynd.
- Nú er $\theta = \dfrac{360^\circ}{n}$ og $\omega = \dfrac{180^\circ - \theta}{2}$.
- Sínus reglan gefur okkur svo að

$$\frac{2R}{\sin\theta} = \frac{r+R}{\sin\omega} \Rightarrow 2R\sin\omega - R\sin\theta = r\sin\theta \Rightarrow R = \frac{r\sin\theta}{2\sin\omega - \sin\theta}$$



Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.

Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- Skilgreinum svo margföldun á $\mathbb{C}$ þannig að fyrir $(a, b), (c, d) \in \mathbb{C}$ þ.a.

$$(a, b) \cdot (c, d) = (ac - bd, ad + bc)$$

Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- Skilgreinum svo margföldun á $\mathbb{C}$ þannig að fyrir $(a,b),(c,d) \in \mathbb{C}$ þ.a.

$$(a,b) \cdot (c,d) = (ac - bd, ad + bc)$$

- Við táknum iðulega $(0,1) \in \mathbb{C}$ með i og $(x,y) \in \mathbb{C}$ með $x + yi$.

Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- Skilgreinum svo margföldun á $\mathbb{C}$ þannig að fyrir $(a,b),(c,d) \in \mathbb{C}$ þ.a.

$$(a,b) \cdot (c,d) = (ac - bd, ad + bc)$$

- Við táknum iðulega $(0,1) \in \mathbb{C}$ með i og $(x,y) \in \mathbb{C}$ með $x + yi$.
- Stök $\mathbb{C}$ kallast *tvinntölur*.

Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- Skilgreinum svo margföldun á $\mathbb{C}$ þannig að fyrir $(a, b), (c, d) \in \mathbb{C}$ þ.a.

$$(a, b) \cdot (c, d) = (ac - bd, ad + bc)$$

- Við táknum iðulega $(0, 1) \in \mathbb{C}$ með i og $(x, y) \in \mathbb{C}$ með $x + yi$.
- Stök $\mathbb{C}$ kallast *tvinntölur*.
- Ef $z = x + yi \in \mathbb{C}$ þá köllum við x *raunhluta* z og y *þverhluta* z.

Tvinntölur

- Skilgreinum mengið $\mathbb{C} := \mathbb{R} \times \mathbb{R}$.
- Skilgreinum svo margföldun á $\mathbb{C}$ þannig að fyrir $(a, b), (c, d) \in \mathbb{C}$ þ.a.

$$(a, b) \cdot (c, d) = (ac - bd, ad + bc)$$

- Við táknum iðulega $(0, 1) \in \mathbb{C}$ með i og $(x, y) \in \mathbb{C}$ með $x + yi$.
- Stök $\mathbb{C}$ kallast *tvinntölur*.
- Ef $z = x + yi \in \mathbb{C}$ þá köllum við x *raunhluta* z og y *þverhluta* z.
- Ef $z = x + yi$ þá er *lengd* z gefin með $|z| = \sqrt{x^2 + y^2}$.
- Ef $z = x + yi$ köllum við $x - yi$ *samoka* z, táknað $\overline{z}$.

Efnisyfirlit

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.
- Þegar leysa þarf þrívíð rúmfræði dæmi er oft góð hugmynd að byrja á að leysa dæmin í tveimur víddum (ef unnt er) og reyna svo að yfirfæra tvívíðu lausnina í þriðju víddina.

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.
- Þegar leysa þarf þrívíð rúmfræði dæmi er oft góð hugmynd að byrja á að leysa dæmin í tveimur víddum (ef unnt er) og reyna svo að yfirfæra tvívíðu lausnina í þriðju víddina.
- Hingað til í námskeiðinu höfum við að mestu fengist við heiltölur og stöku sinnum þurft að vinna með fleytitölur.

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.
- Þegar leysa þarf þrívíð rúmfræði dæmi er oft góð hugmynd að byrja á að leysa dæmin í tveimur víddum (ef unnt er) og reyna svo að yfirfæra tvívíðu lausnina í þriðju víddina.
- Hingað til í námskeiðinu höfum við að mestu fengist við heiltölur og stöku sinnum þurft að vinna með fleytitölur.
- Í rúmfræði er þetta þó öfugt, við vinnum aðallega með fleytitölur og stöku sinnum heiltölur.

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.
- Þegar leysa þarf þrívíð rúmfræði dæmi er oft góð hugmynd að byrja á að leysa dæmin í tveimur víddum (ef unnt er) og reyna svo að yfirfæra tvívíðu lausnina í þriðju víddina.
- Hingað til í námskeiðinu höfum við að mestu fengist við heiltölur og stöku sinnum þurft að vinna með fleytitölur.
- Í rúmfræði er þetta þó öfugt, við vinnum aðallega með fleytitölur og stöku sinnum heiltölur.
- Þegar við notum fleytitölur er mikilvægt að passa að samanburðir er ekki fullkomnir.

Inngangur

- Við munum aðallega fjalla tvívíða rúmfræði í þessum fyrirlestri.
- Þegar leysa þarf þrívíð rúmfræði dæmi er oft góð hugmynd að byrja á að leysa dæmin í tveimur víddum (ef unnt er) og reyna svo að yfirfæra tvívíðu lausnina í þriðju víddina.
- Hingað til í námskeiðinu höfum við að mestu fengist við heiltölur og stöku sinnum þurft að vinna með fleytitölur.
- Í rúmfræði er þetta þó öfugt, við vinnum aðallega með fleytitölur og stöku sinnum heiltölur.
- Þegar við notum fleytitölur er mikilvægt að passa að samanburðir er ekki fullkomnir.
- Við látum því duga að tvær tölur sé *nógu* líkar, í vissum skilningi.

Fleytitölu samanburðir

```c
// daemi um algengan epsilon-slaka
#define EPS 1e-9

int eq(double a, double b)
{
    return fabs(a - b) < EPS;
}

int neq(double a, double b)
{
    return fabs(a - b) >= EPS;
}
```

Punktar

- Það eru tvær algengar leiðir til að útfæra punkta í plani.

Punktar

- Það eru tvær algengar leiðir til að útfæra punkta í plani.
- Augljósari aðferðin er að skilgreina gagnagrind (`struct`) sem geymir tvær fleytitölur.

Punktar

- Það eru tvær algengar leiðir til að útfæra punkta í plani.
- Augljósari aðferðin er að skilgreina gagnagrind (`struct`) sem geymir tvær fleytitölur.
- Hin aðferðin er að nota innbyggða (í flestum málum) tvinntölu gagnatagið.

Punktar

- Það eru tvær algengar leiðir til að útfæra punkta í plani.
- Augljósari aðferðin er að skilgreina gagnagrind (`struct`) sem geymir tvær fleytitölur.
- Hin aðferðin er að nota innbyggða (í flestum málum) tvinntölu gagnatagið.
- Þó þessi aðgerð gæti verið nokkuð heimulleg þá er hún þægileg og fljótleg í útfærslu.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.
- Fallið `abs(p)` skilar fjarlægð p frá $(0, 0)$.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.
- Fallið `abs(p)` skilar fjarlægð p frá $(0, 0)$.
- Fallið `abs(p - q)` skilar fjarlægð milli p og q.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.
- Fallið `abs(p)` skilar fjarlægð p frá $(0, 0)$.
- Fallið `abs(p - q)` skilar fjarlægð milli p og q.
- Fallið `arg(p)` skilar horninu sem p myndar við jákvæða hluta x-ás.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.
- Fallið `abs(p)` skilar fjarlægð p frá $(0, 0)$.
- Fallið `abs(p - q)` skilar fjarlægð milli p og q.
- Fallið `arg(p)` skilar horninu sem p myndar við jákvæða hluta x-ás.
- Fallið `norm(p)` skilar sama og `abs(p)*abs(p)`.

Notkun á `complex` úr C++ í rúmfræði

- Fallið `real(p)` skilar ofanvarpi p á x-ás.
- Fallið `imag(p)` skilar ofanvarpi p á y-ás.
- Fallið `abs(p)` skilar fjarlægð p frá $(0, 0)$.
- Fallið `abs(p - q)` skilar fjarlægð milli p og q.
- Fallið `arg(p)` skilar horninu sem p myndar við jákvæða hluta x-ás.
- Fallið `norm(p)` skilar sama og `abs(p)*abs(p)`.
- Fallið `conj(p)` speglar p um x-ás.

Punktar

```
// 1:
typedef struct
{
    double x, y;
} point;



// 2:
typedef complex<double> point;
```

Sýnidæmi

Þú byrjar í $(0, 0)$ og færð gefnar skipanir. Skapnirnar eru allar einn bókstafur og ein tala. Ef skipunin er `'f' x` gengur þú áfram um x metra, `'b' x` gengur þú aftur á bak um x metra, `'r' x` snýrð þú þér um x gráður til hægri og `'l' x` snýrð þú þér um x gráður til vinstri. Eftir að fylgja öllum þessum skipunum, hversu langt ertu frá $(0, 0)$.

Sýnidæmi

- Ef við erum í $p \in \mathbb{C}$ og viljum taka r metra skref í stefnu θ getum við einfaldlega lagt $r \cdot e^{i\theta}$ við p.

Sýnidæmi

- Ef við erum í $p \in \mathbb{C}$ og viljum taka r metra skref í stefnu θ getum við einfaldlega lagt $r \cdot e^{i\theta}$ við p.
- Hvert við snúum í upphafi skiptir ekki mál því það hefur ekki áhrif á fjarlægðinni til $(0,0)$.

Sýnidæmi

```
int main()
{
    double pi = acos(-1);
    point p(0.0, 0.0);
    double x, r = 0.0;
    int i, n = get_int();
    for (i = 0; i < n; i++)
    {
        char c = getchar();
        x = get_int();

        if (c == 'f')       p = p + x*exp(r*point(0.0, 1.0));
        else if (c == 'b') p = p + x*exp((r + pi)*point(0.0, 1.0));
        else if (c == 'r') r = r - pi*x/180.0;
        else if (c == 'l') r = r + pi*x/180.0;
    }
    printf("%.8f\n", abs(p));
    return 0;
}
```

Línur

- Samkvæmt fyrstu frumsendu rúmfræðinnar skilgreina tveir ólíkir punktar nákvæmlega eina línu.

Línur

- Samkvæmt fyrstu frumsendu rúmfræðinnar skilgreina tveir ólíkir punktar nákvæmlega eina línu.
- Svo við getum einfaldlega sagt að lína sé tveir punktar.

Línur

- Samkvæmt fyrstu frumsendu rúmfræðinnar skilgreina tveir ólíkir punktar nákvæmlega eina línu.
- Svo við getum einfaldlega sagt að lína sé tveir punktar.
- Helsti ókostur þessa aðferðar er að sama línan getur verið skilgreint með mismunandi pörum af puntkum.

Línur

- Samkvæmt fyrstu frumsendu rúmfræðinnar skilgreina tveir ólíkir punktar nákvæmlega eina línu.
- Svo við getum einfaldlega sagt að lína sé tveir punktar.
- Helsti ókostur þessa aðferðar er að sama línan getur verið skilgreint með mismunandi pörum af puntkum.
- Stundum hentar betur að skilgreina línu með skurðpunkt við y-ás og hallatölu.

Línur

- Samkvæmt fyrstu frumsendu rúmfræðinnar skilgreina tveir ólíkir punktar nákvæmlega eina línu.
- Svo við getum einfaldlega sagt að lína sé tveir punktar.
- Helsti ókostur þessa aðferðar er að sama línan getur verið skilgreint með mismunandi pörum af puntkum.
- Stundum hentar betur að skilgreina línu með skurðpunkt við y-ás og hallatölu.
- Þá er einfaldara að bera saman línur en það þarf að höndla sérstaklega línuna í gegnum $(0, 0)$ og $(0, 1)$.

Skurðpunktur lína

- Ef tvær línur eru ekki samsíða þá skerast þær í nákvæmlega einum punkti.

Skurðpunktur lína

- Ef tvær línur eru ekki samsíða þá skerast þær í nákvæmlega einum punkti.
- Við þurfum oft að finna þennan punkt.

Skurðpunktur lína

- Ef tvær línur eru ekki samsíða þá skerast þær í nákvæmlega einum punkti.
- Við þurfum oft að finna þennan punkt.
- Gefum okkur tvær línur $\{(x, y) : ax + by = c\}$ og $\{(x, y) : dx + ey = f\}$.

Skurðpunktur lína

- Ef tvær línur eru ekki samsíða þá skerast þær í nákvæmlega einum punkti.
- Við þurfum oft að finna þennan punkt.
- Gefum okkur tvær línur $\{(x, y) : ax + by = c\}$ og $\{(x, y) : dx + ey = f\}$.
- Gerum ráð fyrir að línurnar séu ekki samsíða.

Skurðpunktur lína

- Ef tvær línur eru ekki samsíða þá skerast þær í nákvæmlega einum punkti.
- Við þurfum oft að finna þennan punkt.
- Gefum okkur tvær línur $\{(x, y) : ax + by = c\}$ og $\{(x, y) : dx + ey = f\}$.
- Gerum ráð fyrir að línurnar séu ekki samsíða.
- Skurðpunkturinn fæst þá greinilega með því að leysa jöfnuhneppið

$$\left(\begin{array}{cc|c} a & b & c \\ d & e & f \end{array}\right)$$

Skurðpunktur lína

$$\begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix}$$

Skurðpunktur lína

$$\begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix}^{-1} \begin{pmatrix} c \\ f \end{pmatrix}$$

Skurðpunktur lína

$$\begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix}^{-1} \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ae - bd} \begin{pmatrix} e & -b \\ -d & a \end{pmatrix} \begin{pmatrix} c \\ f \end{pmatrix}$$

Skurðpunktur lína

$$\begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix}^{-1} \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ae - bd} \begin{pmatrix} e & -b \\ -d & a \end{pmatrix} \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ae - bd} \begin{pmatrix} ce - bf \\ af - cd \end{pmatrix}$$

Skurðpunktur lína

$$\begin{pmatrix} a & b \\ d & e \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ d & e \end{pmatrix}^{-1} \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ae - bd} \begin{pmatrix} e & -b \\ -d & a \end{pmatrix} \begin{pmatrix} c \\ f \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \frac{1}{ae - bd} \begin{pmatrix} ce - bf \\ af - cd \end{pmatrix}$$

$$\Rightarrow \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{ce-bf}{ae-bd} \\ \frac{af-cd}{ae-bd} \end{pmatrix}$$

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.
- Við munum útfæra:

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.
- Við munum útfæra:
    - Fall sem skoðar hvort tvö bil skerist (`bxb`).

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.
- Við munum útfæra:
  - Fall sem skoðar hvort tvö bil skerist (`bxb`).
  - Fall sem skoðar hvort línustrik skerist (`lxl`).

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.
- Við munum útfæra:
    - Fall sem skoðar hvort tvö bil skerist (`bxb`).
    - Fall sem skoðar hvort línustrik skerist (`lxl`).
    - Fall sem finnur stystu fjarlægð punkts og línustriks (`p2l`).

Línustrik

- Af augljósum ástæðum sést að best er að skilgreina línustrik sem par punkta, nánar tiltekið endapunkta línustriksins.
- Markmið okkar í þessum hluta af fyrirlestrinu verður að útfæra fall sem finnur fjarlægð milli línustrika.
- Við munum byrja á að útfæra góð hjálparföll sem nýtast meðal annars í að finna þessa fjarlægð, en eru einnig hentug í öðrum dæmum.
- Við munum útfæra:
  - Fall sem skoðar hvort tvö bil skerist (`bxb`).
  - Fall sem skoðar hvort línustrik skerist (`lxl`).
  - Fall sem finnur stystu fjarlægð punkts og línustriks (`p2l`).
  - Fall sem finnur stystu fjarlægð tveggja línustrika (`l2l`).

Skurður bila

- Þegar við viljum skoða skurð tveggja bila nægir okkur að skoða hvort annar endapunktur bils er í hinu bilinu.

Skurður bila

- Þegar við viljum skoða skurð tveggja bila nægir okkur að skoða hvort annar endapunktur bils er í hinu bilinu.
- ```
  // Skerast [a, b] og [c, d]?
  bool bxb(double a, double b, double c, double d)
  {
      if (a > b) swap(a, b);
      if (c > d) swap(c, d);
      return fmax(a,c) <= fmin(b,d);
  }
  ```
  ```

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.
- Ef þríhyningurinn stikaður með $\langle a, b, c, a \rangle$ hefur öfuga áttun miðað við $\langle a, b, d, a \rangle$ þá liggja punktarnir c og d sitthvoru megin við línustrikið $\langle a, b \rangle$.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.
- Ef þríhyningurinn stikaður með $\langle a, b, c, a \rangle$ hefur öfuga áttun miðað við $\langle a, b, d, a \rangle$ þá liggja punktarnir c og d sitthvoru megin við línustrikið $\langle a, b \rangle$.
- Við þurfum líka að ganga úr skugga um ferhyrningarnir sem endapunktarnir mynda skerast.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.
- Ef þríhyningurinn stikaður með $\langle a, b, c, a \rangle$ hefur öfuga áttun miðað við $\langle a, b, d, a \rangle$ þá liggja punktarnir c og d sitthvoru megin við línustrikið $\langle a, b \rangle$.
- Við þurfum líka að ganga úr skugga um ferhyrningarnir sem endapunktarnir mynda skerast.
- Það eru leiðinleg sértilfelli þegar þrír af endapunktunum liggja á sömulínunni.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.
- Ef þríhyningurinn stikaður með $\langle a, b, c, a \rangle$ hefur öfuga áttun miðað við $\langle a, b, d, a \rangle$ þá liggja punktarnir c og d sitthvoru megin við línustrikið $\langle a, b \rangle$.
- Við þurfum líka að ganga úr skugga um ferhyrningarnir sem endapunktarnir mynda skerast.
- Það eru leiðinleg sértilfelli þegar þrír af endapunktunum liggja á sömulínunni.
- Ég mun eftirláta ykkur að laga þetta sértilfelli, því þar sem við höfum aðallega áhuga á að finna fjarlægð línubila nægir að segja að línustrikin skerist ekki í þessu sértilfelli.

Skurður tveggja línustrika

- Þessi glæra gæti verið nokkuð dularfull en hún mun verða skýrari seinna í þessum fyrirlestri.
- Við munum skoða áttunina á þríhyrningunum sem við getum myndað með endapunktum línustrikanna.
- Látum a, b, c, d vera endapunkta línustrikanna.
- Ef þríhyningurinn stikaður með $\langle a, b, c, a\rangle$ hefur öfuga áttun miðað við $\langle a, b, d, a\rangle$ þá liggja punktarnir c og d sitthvoru megin við línustrikið $\langle a, b\rangle$.
- Við þurfum líka að ganga úr skugga um ferhyrningarnir sem endapunktarnir mynda skerast.
- Það eru leiðinleg sértilfelli þegar þrír af endapunktunum liggja á sömulínunni.
- Ég mun eftirláta ykkur að laga þetta sértilfelli, því þar sem við höfum aðallega áhuga á að finna fjarlægð línubila nægir að segja að línustrikin skerist ekki í þessu sértilfelli.
- Þetta mun skýrast betur á eftir.

Skurður tveggja línustrika

```
// Skilar attun á þrihyrningnum stikuðum með <a, b, c, a>
int sgnarea(point a, point b, point c)
{
    double f = real(b - a)*imag(c - a) - imag(b - a)*real(c - a);
    if (fabs(f) < EPS) return 0;
    if (f < EPS) return -1;
    return 1;
}

// Skerast <a, b> og <c, d>?
int lxl(point a, point b, point c, point d)
{
    int a1 = sgnarea(a, b, c), a2 = sgnarea(a, b, d),
        a3 = sgnarea(c, d, a), a4 = sgnarea(c, d, b);
    if (a1*a2*a3*a4 == 0) return 0; // ATH: virkar almennt ekki
    if (a1*a2 != -1 || a3*a4 != -1) return 0;
    return bxb(real(a), real(b), real(c), real(d))
        && bxb(imag(a), imag(b), imag(c), imag(d));
}
```

Fjarlægð punkts og línustriks

- Hér hefst fjörið.

Fjarlægð punkts og línustriks

- Hér hefst fjörið.
- Látum línustrikið vera $\langle (x_0, y_0), (x_1, y_1) \rangle$ og punktinn (x, y).

Fjarlægð punkts og línustriks

- Hér hefst fjörið.
- Látum línustrikið vera $\langle (x_0, y_0), (x_1, y_1) \rangle$ og punktinn (x, y).
- Við getum þá stikað línustrikið með
  $l(t) = (x_0 + t \cdot (x_1 - x_0), y_0 + t \cdot (y_1 - y_0)), t \in [0, 1]$.

Fjarlægð punkts og línustriks

- Hér hefst fjörið.
- Látum línustrikið vera $\langle (x_0, y_0), (x_1, y_1) \rangle$ og punktinn (x, y).
- Við getum þá stikað línustrikið með $l(t) = (x_0 + t \cdot (x_1 - x_0), y_0 + t \cdot (y_1 - y_0)), t \in [0, 1]$.
- Látum d tákna Evklíðsku firðina. Við munum nú lágmarka d^2 til að auðvelda deildunina.

Fjarlægð punkts og línustriks

- Hér hefst fjörið.
- Látum línustrikið vera $\langle(x_0, y_0), (x_1, y_1)\rangle$ og punktinn (x, y).
- Við getum þá stikað línustrikið með $l(t) = (x_0 + t \cdot (x_1 - x_0), y_0 + t \cdot (y_1 - y_0)), t \in [0, 1]$.
- Látum d tákna Evklíðsku firðina. Við munum nú lágmarka d^2 til að auðvelda deildunina.
- Látum $f(t) = d(l(t), (x, y))^2$.

Fjarlægð punkts og línustriks

- Hér hefst fjörið.
- Látum línustrikið vera $\langle (x_0, y_0), (x_1, y_1) \rangle$ og punktinn (x, y).
- Við getum þá stikað línustrikið með $l(t) = (x_0 + t \cdot (x_1 - x_0), y_0 + t \cdot (y_1 - y_0)), t \in [0, 1]$.
- Látum d tákna Evklíðsku firðina. Við munum nú lágmarka d^2 til að auðvelda deildunina.
- Látum $f(t) = d(l(t), (x, y))^2$.
- Við fáum enn fremur að

$$\begin{aligned}
f(t) =& (x_0 + t \cdot (x_1 - x_0) - x)^2 + (y_0 + t \cdot (y_1 - y_0) - y)^2 \\
=& x_0^2 + t^2 (x_1 - x_0)^2 + x^2 + 2x_0 t (x_1 - x_0) - 2xx_0 - 2tx(x_1 - x_0) \\
+& y_0^2 + t^2 (y_1 - y_0)^2 + y^2 + 2y_0 t (y_1 - y_0) - 2yy_0 - 2ty(y_1 - y_0).
\end{aligned}$$

Fjarlægð punkts og línustriks

- Deildum nú f og fáum

$$\begin{aligned} f'(t) = & 2t(x_1 - x_0)^2 + 2x_0(x_1 - x_0) - 2x(x_1 - x_0) \\ & + 2t(y_1 - y_0)^2 + 2y_0(y_1 - y_0) - 2y(y_1 - y_0). \end{aligned}$$

Fjarlægð punkts og línustriks

- Deildum nú f og fáum

$$\begin{aligned} f'(t) =& 2t(x_1 - x_0)^2 + 2x_0(x_1 - x_0) - 2x(x_1 - x_0) \\ &+ 2t(y_1 - y_0)^2 + 2y_0(y_1 - y_0) - 2y(y_1 - y_0). \end{aligned}$$

- Metum nú í 0 óg fáum

$$\begin{aligned} & f'(t_0) = 0 \\ \Rightarrow & t_0(x_1 - x_0)^2 + x_0(x_1 - x_0) - x(x_1 - x_0) \\ & + t_0(y_1 - y_0)^2 + y_0(y_1 - y_0) - y(y_1 - y_0) = 0 \\ \Rightarrow & t_0 = \frac{x(x_1 - x_0) + y(y_1 - y_0) - x_0(x_1 - x_0) - y_0(y_1 - y_0)}{(y_1 - y_0)^2 + (x_1 - x_0)^2} \\ \Rightarrow & t_0 = \frac{(x - x_0)(x_1 - x_0) + (y - y_0)(y_1 - y_0)}{(y_1 - y_0)^2 + (x_1 - x_0)^2}. \end{aligned}$$

Fjarlægð punkts og línustriks

- Við erum nú búin að finna þann punkt á línunni gegnum (x_0, y_0) og (x_1, y_1) sem er næstur (x, y).

Fjarlægð punkts og línustriks

- Við erum nú búin að finna þann punkt á línunni gegnum (x_0, y_0) og (x_1, y_1) sem er næstur (x, y).
- Þar sem línustrikið er stikað af t þegar $t \in [0, 1]$ þá er þessi punktur á línustrikinu ef $t_0 \in [0, 1]$.

Fjarlægð punkts og línustriks

- Við erum nú búin að finna þann punkt á línunni gegnum (x_0, y_0) og (x_1, y_1) sem er næstur (x, y).
- Þar sem línustrikið er stikað af t þegar $t \in [0, 1]$ þá er þessi punktur á línustrikinu ef $t_0 \in [0, 1]$.
- Ef svo er ekki nægir okkur að skoða fjarlægð (x, y) til endapunktana (x_0, y_0) og (x_1, y_1).

Fjarlægð punkts og línustriks

- Við erum nú búin að finna þann punkt á línunni gegnum (x_0, y_0) og (x_1, y_1) sem er næstur (x, y).
- Þar sem línustrikið er stikað af t þegar $t \in [0, 1]$ þá er þessi punktur á línustrikinu ef $t_0 \in [0, 1]$.
- Ef svo er ekki nægir okkur að skoða fjarlægð (x, y) til endapunktana (x_0, y_0) og (x_1, y_1).
- ```
  double p2l(point p, point l1, point l2)
  {
      double t = (real(l2 - l1)*real(p - l1) + imag(l2 - l1)*imag(p - l1))
          /norm(l2 - l1);
      if (t > 0.0 && t < 1.0) return abs(l1 + t*(l2 - l1) - p);
      return fmin(abs(l1 - p), abs(l2 - p));
  }
  ```
  ```

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.
- Þá er bersýnilega fjarlægð línustrikana minnst á milli a og $\langle c, d\rangle$, b og $\langle c, d\rangle$, c og $\langle a, b\rangle$ eða d og $\langle a, b\rangle$.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.
- Þá er bersýnilega fjarlægð línustrikana minnst á milli a og $\langle c, d\rangle$, b og $\langle c, d\rangle$, c og $\langle a, b\rangle$ eða d og $\langle a, b\rangle$.
- Ég hef ekki minnst á það hingað til, en við höfum gert ráð fyrir að endapunktar línustrikana séu mismunandi.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.
- Þá er bersýnilega fjarlægð línustrikana minnst á milli a og $\langle c, d\rangle$, b og $\langle c, d\rangle$, c og $\langle a, b\rangle$ eða d og $\langle a, b\rangle$.
- Ég hef ekki minnst á það hingað til, en við höfum gert ráð fyrir að endapunktar línustrikana séu mismunandi.
- Ef $a = b$ og $c = d$ þá er fjarlægð línustrikana fjarlægðin milli a og c.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.
- Þá er bersýnilega fjarlægð línustrikana minnst á milli a og $\langle c, d\rangle$, b og $\langle c, d\rangle$, c og $\langle a, b\rangle$ eða d og $\langle a, b\rangle$.
- Ég hef ekki minnst á það hingað til, en við höfum gert ráð fyrir að endapunktar línustrikana séu mismunandi.
- Ef $a = b$ og $c = d$ þá er fjarlægð línustrikana fjarlægðin milli a og c.
- Ef $a = b$ og $c \neq d$ þá er fjarlægð línustrikana fjarlægðin milli a og $\langle c, d\rangle$.

Fjarlægð milli tveggja línustrika

- Gefum okkur línustrikin $\langle a, b\rangle$ og $\langle c, d\rangle$.
- Ef þau skerast þá er fjarlægðin á milli þeirra bersýnilega 0.
- Gerum því ráð fyrir að þau skerist ekki.
- Þá er bersýnilega fjarlægð línustrikana minnst á milli a og $\langle c, d\rangle$, b og $\langle c, d\rangle$, c og $\langle a, b\rangle$ eða d og $\langle a, b\rangle$.
- Ég hef ekki minnst á það hingað til, en við höfum gert ráð fyrir að endapunktar línustrikana séu mismunandi.
- Ef $a = b$ og $c = d$ þá er fjarlægð línustrikana fjarlægðin milli a og c.
- Ef $a = b$ og $c \neq d$ þá er fjarlægð línustrikana fjarlægðin milli a og $\langle c, d\rangle$.
- Ef $a \neq b$ og $c = d$ þá er fjarlægð línustrikana fjarlægðin milli c og $\langle a, b\rangle$.

Fjarlægð milli tveggja línustrika

```
double l2l(point a1, point a2, point b1, point b2)
{
    if (fabs(abs(a1 - a2)) < EPS && fabs(abs(b1 - b2)) < EPS)
        return abs(a1 - b1);
    if (fabs(abs(a1 - a2)) < EPS) return p2l(a1, b1, b2);
    if (fabs(abs(b1 - b2)) < EPS) return p2l(b1, a1, a2);
    if (lxl(a1, a2, b1, b2)) return 0.0;
    return fmin(fmin(p2l(a1, b1, b2), p2l(a2, b1, b2)),
                fmin(p2l(b1, a1, a2), p2l(b2, a1, a2)));
}
```

Hringir

- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.

Hringir

- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.
- Það er gott að vita að unnt er að ákvarð hring útfrá:

Hringir

- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.
- Það er gott að vita að unnt er að ákvarð hring útfrá:
  - Miðju og geisla.

Hringir

- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.
- Það er gott að vita að unnt er að ákvarð hring útfrá:
  - Miðju og geisla.
  - Miðju og og punkti á jaðri hringsins.

Hringir

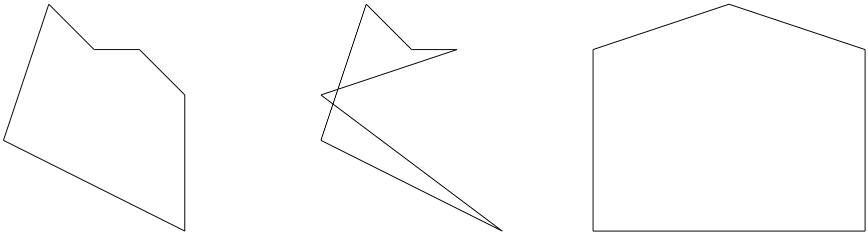
- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.
- Það er gott að vita að unnt er að ákvarð hring útfrá:
    - Miðju og geisla.
    - Miðju og og punkti á jaðri hringsins.
    - Þremur punktum á jaðri hringsins.

Hringir

- Til að útfæra hringi geymum við iðulega miðpunkt hringsins og geisla hans.
- Það er gott að vita að unnt er að ákvarð hring útfrá:
  - Miðju og geisla.
  - Miðju og og punkti á jaðri hringsins.
  - Þremur punktum á jaðri hringsins.
  - Tveimur punktum á jaðri hringsins og geisla (hér er þó tveir mögulegir hringir).

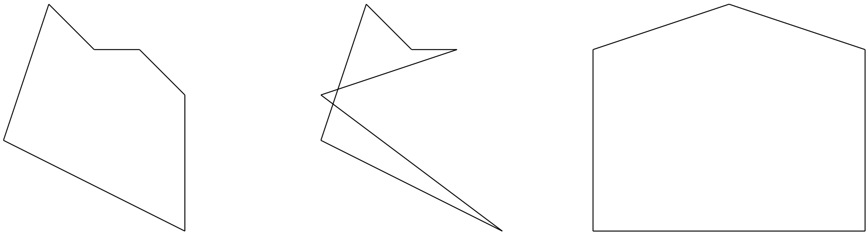
Marghyrningar

- *Marghyrningur* er samfelldur, lokaður ferill í plani sem samanstendur af beinum línustrikum.

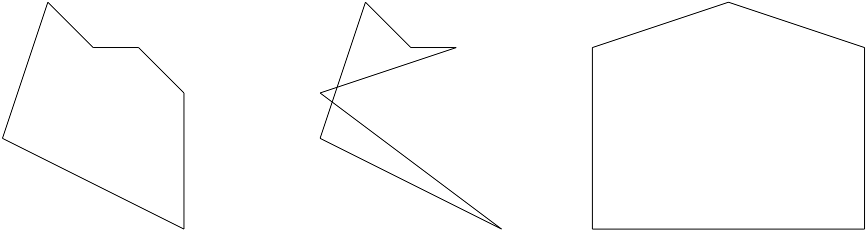


Marghyrningar

- *Marghyrningur* er samfelldur, lokaður ferill í plani sem samanstendur af beinum línustrikum.
- Ef ferillinn er einfaldur þá kallast marghyrningurinn *einfaldur*.



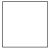
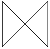
- *Marghyrningur* er samfelldur, lokaður ferill í plani sem samanstendur af beinum línustrikum.
- Ef ferillinn er einfaldur þá kallast marghyrningurinn *einfaldur*.
- Marghyrningur eru sagður vera *kúptur* ef sérhver beina lína dregin gegnum marghyrninginn sker mest tvo punkta á marghyrningnum.



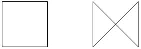
Framsetning marghyrninga

- Þegar við viljum tákna marghyrning í tölvu notum við einfaldlega hornpunkta hans.

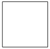
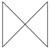
Framsetning marghyrninga

- Þegar við viljum tákna marghyrning í tölvu notum við einfaldlega hornpunkta hans.
- Röð punktanna skiptir máli.  

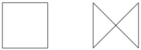
Framsetning marghyrninga

- Þegar við viljum tákna marghyrning í tölvu notum við einfaldlega hornpunkta hans.
- Röð punktanna skiptir máli. 
- Til þæginda geymum við einn punkt tvisvar, nánar tiltekið er fremsti og aftasti punkturinn eins.

Framsetning marghyrninga

- Þegar við viljum tákna marghyrning í tölvu notum við einfaldlega hornpunkta hans.
- Röð punktanna skiptir máli.  
- Til þæginda geymum við einn punkt tvisvar, nánar tiltekið er fremsti og aftasti punkturinn eins.
- Þetta er því við höfum oft meira áhuga á línustrikunum milli hornpuntkanna heldur en hornpunktunum sjálfum.

Framsetning marghyrninga

- Þegar við viljum tákna marghyrning í tölvu notum við einfaldlega hornpunkta hans.
- Röð punktanna skiptir máli. 
- Til þæginda geymum við einn punkt tvisvar, nánar tiltekið er fremsti og aftasti punkturinn eins.
- Þetta er því við höfum oft meira áhuga á línustrikunum milli hornpuntkanna heldur en hornpunktunum sjálfum.
-
```
typedef vector<point> polygon;
```

Nokkur atriði um marghyrninga

- Þar sem marghyrningar er mjög vinsælir í keppnum munum við fara í nokkur atriði sem er gott að kunna.

Ummál marghyrnings

- Ummála marghyrnings er einfalt að reikna í línulegum tíma.

Ummál marghyrnings

- Ummála marghyrnings er einfalt að reikna í línulegum tíma.
- Maður leggur einfaldlega saman allar hliðarlengdirnar.

Ummál marghyrnings

```
// p[0] == p[n - 1]
double ummal(polygon p)
{
    int i;
    double r = 0.0;
    for (i = 0; i < p.size() - 1; i++)
    {
        r = r + abs(p[i] - p[i + 1]);
    }
    return r;
}
```

Flatarmál marghyrnings

- Ummál marghyrninga er þó ekki jafnt algengt í keppnum og flatarmál marghyrninga.

Flatarmál marghyrnings

- Ummál marghyrninga er þó ekki jafnt algengt í keppnum og flatarmál marghyrninga.
- Það er einnig auðvelt að reikna flatarmálið í línulegum tíma, þó það sé ekki endilega augljóst að þetta skili flatarmálinu.

Flatarmál marghyrnings

- Ummál marghyrninga er þó ekki jafnt algengt í keppnum og flatarmál marghyrninga.
- Það er einnig auðvelt að reikna flatarmálið í línulegum tíma, þó það sé ekki endilega augljóst að þetta skili flatarmálinu.
- Fyrir áhugasama er hægt að nota setningu Green til að leiða út eftirfarandi forritsbút.

Flatarmál marghyrnings

- ```
  double flatarmal(polygon &p)
  {
      int i;
      double r = 0.0;
      for (i = 0; i < p.size() - 1; i++)
      {
          r = r + real(p[i])*imag(p[i + 1]) - real(p[i + 1])*imag(p[i]);
      }
      return fabs(0.5*r);
  }
  ```
  ```

Flatarmál marghyrnings

- ```
  double flatarmal(polygon &p)
  {
      int i;
      double r = 0.0;
      for (i = 0; i < p.size() - 1; i++)
      {
          r = r + real(p[i])*imag(p[i + 1]) - real(p[i + 1])*imag(p[i]);
      }
      return fabs(0.5*r);
  }
  ```

- Við þurfum að skila tölugildinu því við getum fengið neikvætt flatarmál.
  ```

Flatarmál marghyrnings

- ```
  double flatarmal(polygon &p)
  {
      int i;
      double r = 0.0;
      for (i = 0; i < p.size() - 1; i++)
      {
          r = r + real(p[i])*imag(p[i + 1]) - real(p[i + 1])*imag(p[i]);
      }
      return fabs(0.5*r);
  }
  ```

- Við þurfum að skila tölugildinu því við getum fengið neikvætt flatarmál.
- Neikvætt flatarmál hljómar kannski furðulega en það fellur eðlilega úr sönnun summunar ef notast er við setningu Green.
  ```

Flatarmál marghyrnings

- ```
  double flatarmal(polygon &p)
  {
      int i;
      double r = 0.0;
      for (i = 0; i < p.size() - 1; i++)
      {
          r = r + real(p[i])*imag(p[i + 1]) - real(p[i + 1])*imag(p[i]);
      }
      return fabs(0.5*r);
  }
  ```
- Við þurfum að skila tölugildinu því við getum fengið neikvætt flatarmál.
- Neikvætt flatarmál hljómar kannski furðulega en það fellur eðlilega úr sönnun summunar ef notast er við setningu Green.
- Til að nota hana þarf að reikna ferilheildi og útkoman úr ferilheildum skiptir um formerki þegar breytt er um átt stikunar ferilsins.
  ```

Flatarmál marghyrnings

- ```
  double flatarmal(polygon &p)
  {
      int i;
      double r = 0.0;
      for (i = 0; i < p.size() - 1; i++)
      {
          r = r + real(p[i])*imag(p[i + 1]) - real(p[i + 1])*imag(p[i]);
      }
      return fabs(0.5*r);
  }
  ```

- Við þurfum að skila tölugildinu því við getum fengið neikvætt flatarmál.
- Neikvætt flatarmál hljómar kannski furðulega en það fellur eðlilega úr sönnun summunar ef notast er við setningu Green.
- Til að nota hana þarf að reikna ferilheildi og útkoman úr ferilheildum skiptir um formerki þegar breytt er um átt stikunar ferilsins.
- Þetta þýðir að formerki `r` eftir forlykkjuna er jákvætt ef punktar `p` eru gefnir rangsælis og neikvætt ef þeir eru gefnir réttsælis.
  ```

Punktur í marghyrning

- Að ákvarða hvort punktur sé inni í marghyrning (e. the point in polygon problem) er algent undirvandamál í rúmfræði dæmum.

Punktur í marghyrning

- Að ákvarða hvort punktur sé inni í marghyrning (e. the point in polygon problem) er algent undirvandamál í rúmfræði dæmum.
- Aðallega er gengist við tvær aðferðir til að leysa slík dæmi, sú fyrri er að nota geislarakningu (e. raytracing) og hin er að reikna summu aðliggjandi horna marghyrningsins miðað við punktinn.

Geislarakning

- Við getum dregið geisla frá punktinum sem við viljum skoða í einhverja átt og talið hversu oft við skerum jaðar marghyrningsins.

Geislarakning

- Við getum dregið geisla frá punktinum sem við viljum skoða í einhverja átt og talið hversu oft við skerum jaðar marghyrningsins.
- Ef við erum fyrir utan marghyrninginn og skerum jaðarinn erum við inni í honum, en ef við erum fyrir utan og skerum jaðarinn erum við fyrir innan (þetta er í raun skilgreining á því hvenær geislinn *sker* marghyrninginn).

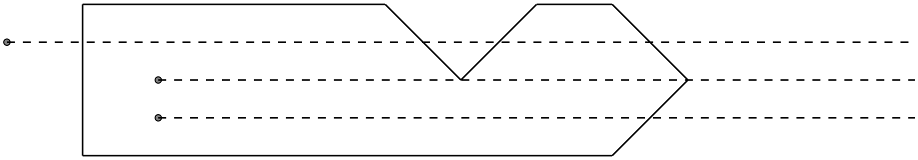
Geislarakning

- Við getum dregið geisla frá punktinum sem við viljum skoða í einhverja átt og talið hversu oft við skerum jaðar marghyrningsins.
- Ef við erum fyrir utan marghyrninginn og skerum jaðarinn erum við inni í honum, en ef við erum fyrir utan og skerum jaðarinn erum við fyrir innan (þetta er í raun skilgreining á því hvenær geislinn *sker* marghyrninginn).
- Svo ef við skerum jaðarinn slétt tölu sinnum er punkturinn fyrir innan, og annars fyrir utan (Setning Jordan).

Geislarakning

- Við getum dregið geisla frá punktinum sem við viljum skoða í einhverja átt og talið hversu oft við skerum jaðar marghyrningsins.
- Ef við erum fyrir utan marghyrninginn og skerum jaðarinn erum við inni í honum, en ef við erum fyrir utan og skerum jaðarinn erum við fyrir innan (þetta er í raun skilgreining á því hvenær geislinn *sker* marghyrninginn).
- Svo ef við skerum jaðarinn slétt tölu sinnum er punkturinn fyrir innan, og annars fyrir utan (Setning Jordan).
- Við getum látið geislann vera línustrik, nógu langt til að vera út fyrir marghyrninginn, og notað síðan `lxl` til að ákvarða í línulegum tíma hversu oft geislinn sker marghyrninginn.

Geislarakning



Geislarakning

- Þessi aðferð er með aragrúa af sértilfellum sem gerir þessa aðferð nokkuð óþægilega í útfærslu.

Geislarakning

- Þessi aðferð er með aragrúa af sértilfellum sem gerir þessa aðferð nokkuð óþægilega í útfærslu.
- Öll sértilfellin eiga það sameiginlegt að vera þegar geislinn sker endapunkta línustrika marghyrningsins.

Geislarakning

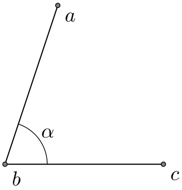
- Þessi aðferð er með aragrúa af sértilfellum sem gerir þessa aðferð nokkuð óþægilega í útfærslu.
- Öll sértilfellin eiga það sameiginlegt að vera þegar geislinn sker endapunkta línustrika marghyrningsins.
- Ef marghyrningurinn er kúptur er nokkuð auðvelt að eiga við þessi sértilfelli, en það gildir ekki í flestum dæmum.

Geislarakning

- Þessi aðferð er með aragrúa af sértilfellum sem gerir þessa aðferð nokkuð óþægilega í útfærslu.
- Öll sértilfellin eiga það sameiginlegt að vera þegar geislinn sker endapunkta línustrika marghyrningsins.
- Ef marghyrningurinn er kúptur er nokkuð auðvelt að eiga við þessi sértilfelli, en það gildir ekki í flestum dæmum.
- Þessi aðferð verður því ekki úrfærð hér.

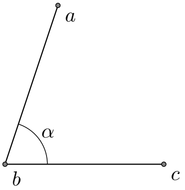
Afstæð hornasumma

- Látum p_i, $i < n$ tákna hornpunkta marghyrnings, p einhvern punkt, $\alpha(a, b, c)$ vera hornið milli sem punktarnir a, b og c mynda og $\beta(a, b, c)$ vera 1 ef brotna línustrikið $\langle a, b, c \rangle$ „beygir" til vinstri en -1 annars.



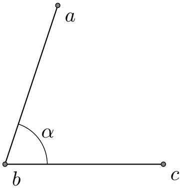
Afstæð hornasumma

- Látum p_i, $i < n$ tákna hornpunkta marghyrnings, p einhvern punkt, $\alpha(a, b, c)$ vera hornið milli sem punktarnir a, b og c mynda og $\beta(a, b, c)$ vera 1 ef brotna línustrikið $\langle a, b, c \rangle$ „beygir" til vinstri en -1 annars.
- *Afstæð hornsumma marghyrnings með tilliti til punkts* q er $\sum_{i=0}^{n} \beta(q, p_i, p_{i+1})\alpha(p_i, q, p_{i+1})$.



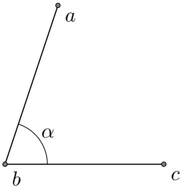
Afstæð hornasumma

- Látum p_i, $i < n$ tákna hornpunkta marghyrnings, p einhvern punkt, $\alpha(a, b, c)$ vera hornið milli sem punktarnir a, b og c mynda og $\beta(a, b, c)$ vera 1 ef brotna línustrikið $\langle a, b, c\rangle$ „beygir" til vinstri en -1 annars.
- *Afstæð hornsumma marghyrnings með tilliti til punkts* q er $\sum_{i=0}^{n} \beta(q, p_i, p_{i+1})\alpha(p_i, q, p_{i+1})$.
- Ef q er inni í marghyrningnum þá er þessi summa bersýnilega 2π.

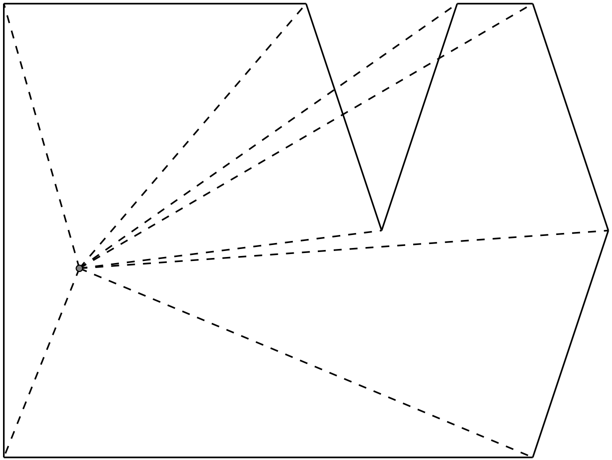


Afstæð hornasumma

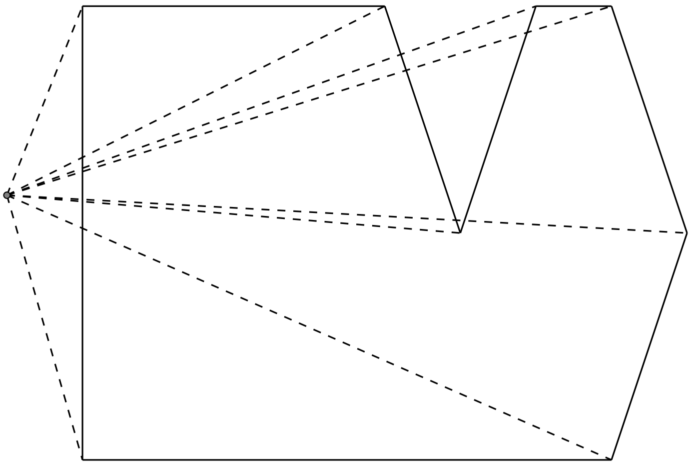
- Látum p_i, $i < n$ tákna hornpunkta marghyrnings, p einhvern punkt, $\alpha(a, b, c)$ vera hornið milli sem punktarnir a, b og c mynda og $\beta(a, b, c)$ vera 1 ef brotna línustrikið $\langle a, b, c\rangle$ „beygir" til vinstri en -1 annars.
- *Afstæð hornsumma marghyrnings með tilliti til punkts* q er $\sum_{i=0}^{n} \beta(q, p_i, p_{i+1})\alpha(p_i, q, p_{i+1})$.
- Ef q er inni í marghyrningnum þá er þessi summa bersýnilega 2π.
- Ef q er fyrir utan marghyrninginn þá verður summan hins vegar 0.



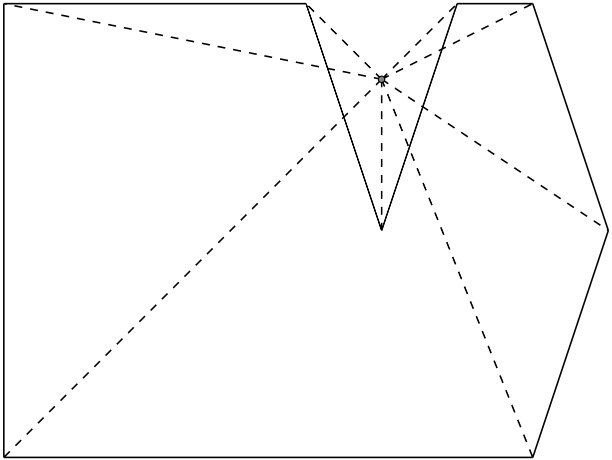
Afstæð hornasumma



Afstæð hornasumma



Afstæð hornasumma



Afstæð hornasumma

```
// alpha í glaerunum
double angle(point a, point o, point b)
{
    double r = fabs(arg(a - o) - arg(b - o));
    return r < M_PI ? r : 2*M_PI - r;
}

int beta(point a, point b, point c)
{
    return real(b - a)*imag(c - a) - imag(b - a)*real(c - a) > 0.0 ? 1 : -1;
}

int is_in(polygon& p, point q)
{
    int i;
    double s = 0.0;
    for (i = 0; i < p.size() - 1; i++)
        s = s + beta(q, p[i], p[i + 1])*angle(p[i], q, p[i + 1]);
    return (fabs(s) > M_PI ? 1 : 0);
}
```

Kútpur hjúpur punktasafns

- *Kúptur hjúpur punktasafns* er minnsti kúpti marghyrningur sem inniheldur all punkta punktasafnsins.

Kútpur hjúpur punktasafns

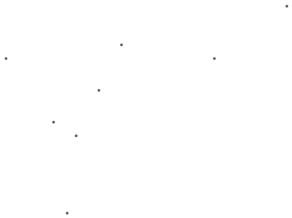
- *Kúptur hjúpur punktasafns* er minnsti kúpti marghyrningur sem inniheldur all punkta punktasafnsins.
- Maður getur ímyndað sér að maður taki teygju og strekki hana yfir punkta safnið og sleppi henni svo.

Kútpur hjúpur punktasafns

- *Kúptur hjúpur punktasafns* er minnsti kúpti marghyrningur sem inniheldur all punkta punktasafnsins.
- Maður getur ímyndað sér að maður taki teygju og strekki hana yfir punkta safnið og sleppi henni svo.
- Við munum nota aðferð sem kallast *Graham's scan* til að finna kúpta hjúp punktasafns.

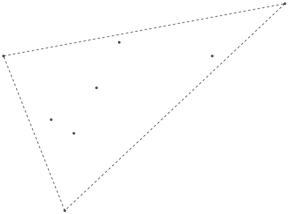
Kútpur hjúpur punktasafns

- *Kúptur hjúpur punktasafns* er minnsti kúpti marghyrningur sem inniheldur all punkta punktasafnsins.
- Maður getur ímyndað sér að maður taki teygju og strekki hana yfir punkta safnið og sleppi henni svo.
- Við munum nota aðferð sem kallast *Graham's scan* til að finna kúpta hjúp punktasafns.



Kútpur hjúpur punktasafns

- *Kúptur hjúpur punktasafns* er minnsti kúpti marghyrningur sem inniheldur all punkta punktasafnsins.
- Maður getur ímyndað sér að maður taki teygju og strekki hana yfir punkta safnið og sleppi henni svo.
- Við munum nota aðferð sem kallast *Graham's scan* til að finna kúpta hjúp punktasafns.



Graham's Scan

- Við byrjum á að velja vendipunkt, yfirleitt látin vera punkturinn neðst til vinstri.

Graham's Scan

- Við byrjum á að velja vendipunkt, yfirleitt látin vera punkturinn neðst til vinstri.
- Við látum hann fremst í safnið og röðum svo restin miðið við hornið sem þeir mynda við vendipunktinn.

Graham's Scan

- Við byrjum á að velja vendipunkt, yfirleitt látin vera punkturinn neðst til vinstri.
- Við látum hann fremst í safnið og röðum svo restin miðið við hornið sem þeir mynda við vendipunktinn.
- Við gefum okkur svo hlaða og látum aftasta, fremsta og næst fremsta punktinn á hlaðann.

Graham's Scan

- Við byrjum á að velja vendipunkt, yfirleitt látin vera punkturinn neðst til vinstri.
- Við látum hann fremst í safnið og röðum svo restin miðið við hornið sem þeir mynda við vendipunktinn.
- Við gefum okkur svo hlaða og látum aftasta, fremsta og næst fremsta punktinn á hlaðann.
- Við göngum síðan í gegnum raðaða punkta safnið okkur og fyrir hvert stak fjarlægjum við ofan af hlaðanum á meðan efstu tvö stökin á hlaðanum og stakið sem við erum á í listanum mynda hægri beygju. Þegar þau mynda vinstri beygju bætum við stakinu úr safninu á hlaðan.

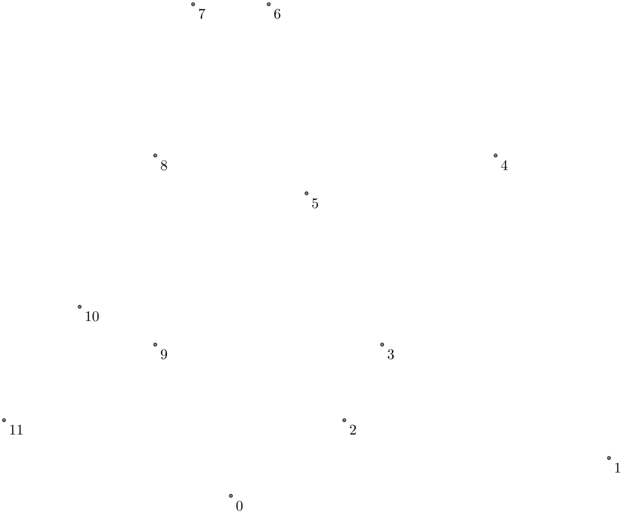
Graham's Scan

- Við byrjum á að velja vendipunkt, yfirleitt látin vera punkturinn neðst til vinstri.
- Við látum hann fremst í safnið og röðum svo restin miðið við hornið sem þeir mynda við vendipunktinn.
- Við gefum okkur svo hlaða og látum aftasta, fremsta og næst fremsta punktinn á hlaðann.
- Við göngum síðan í gegnum raðaða punkta safnið okkur og fyrir hvert stak fjarlægjum við ofan af hlaðanum á meðan efstu tvö stökin á hlaðanum og stakið sem við erum á í listanum mynda hægri beygju. Þegar þau mynda vinstri beygju bætum við stakinu úr safninu á hlaðan.
- Þegar við er búin að fara í gegnum allt safnið er hlaðinn kúpti hjúpurinn.

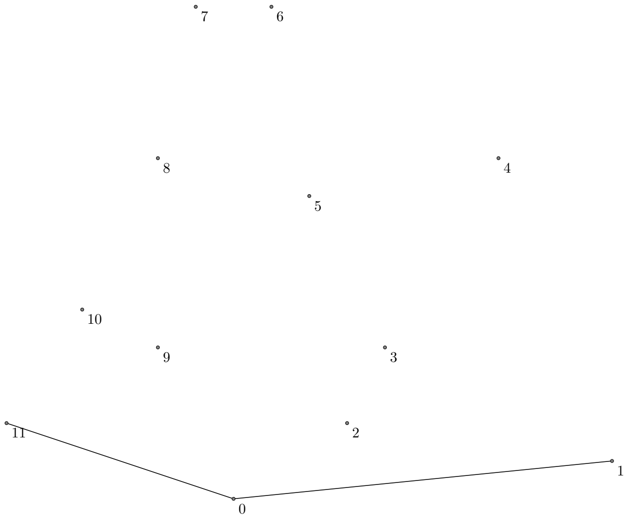
Kútpur hjúpur punktasafns

Kútpur hjúpur punktasafns

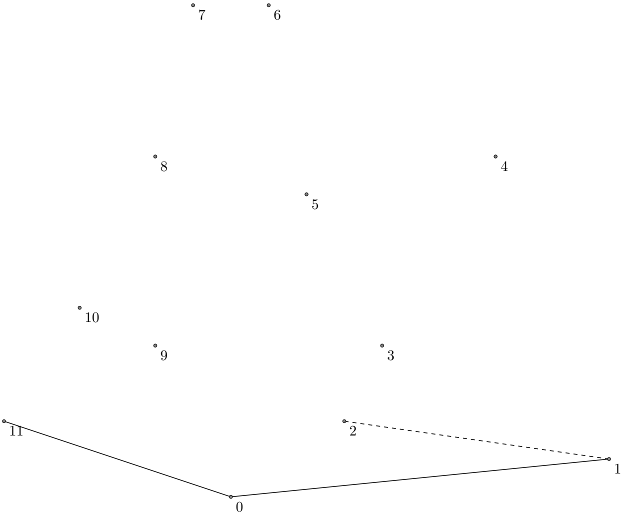
Kútpur hjúpur punktasafns



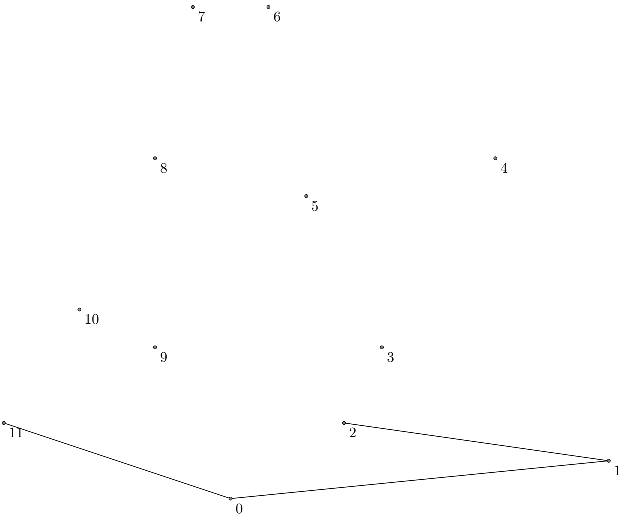
Kútpur hjúpur punktasafns



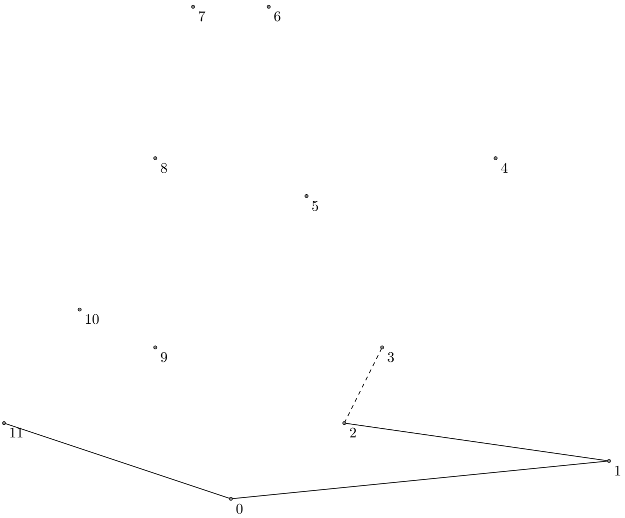
Kútpur hjúpur punktasafns



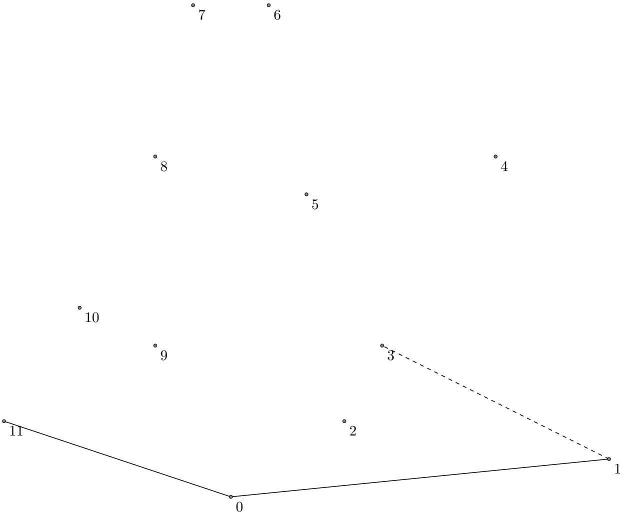
Kútpur hjúpur punktasafns



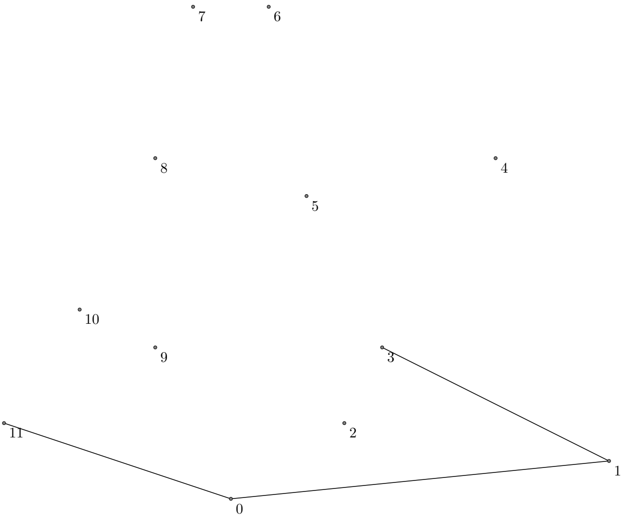
Kútpur hjúpur punktasafns



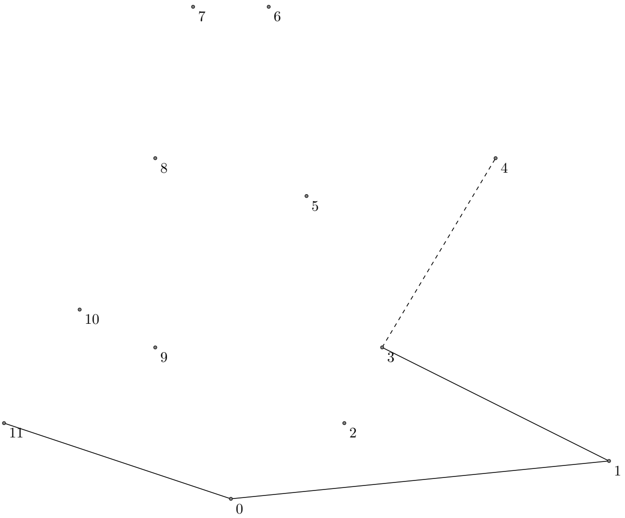
Kútpur hjúpur punktasafns



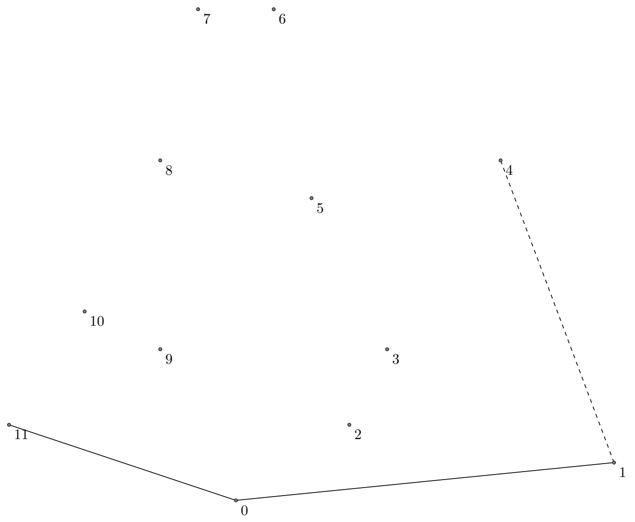
Kútpur hjúpur punktasafns



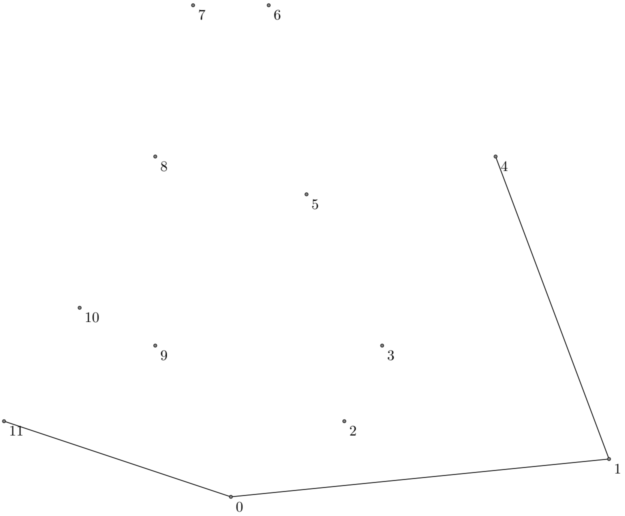
Kútpur hjúpur punktasafns



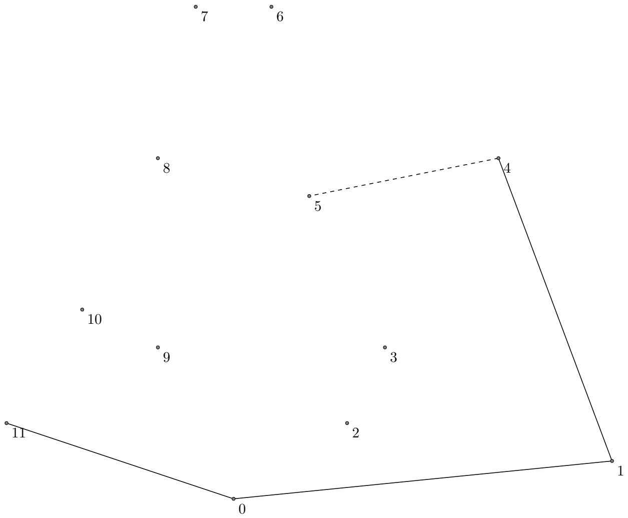
Kútpur hjúpur punktasafns



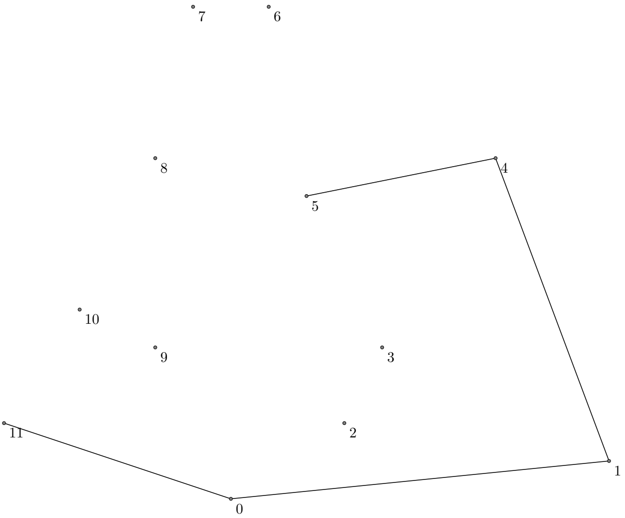
Kútpur hjúpur punktasafns



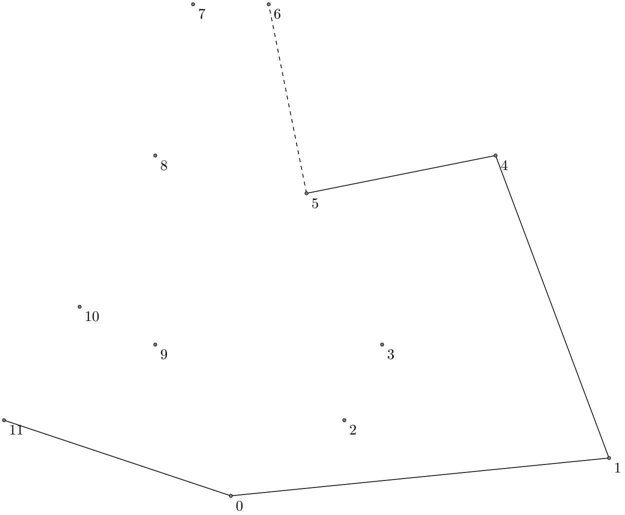
Kútpur hjúpur punktasafns



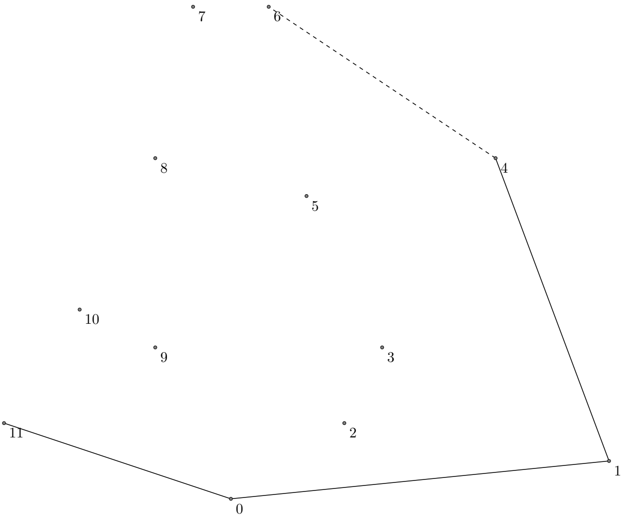
Kútpur hjúpur punktasafns



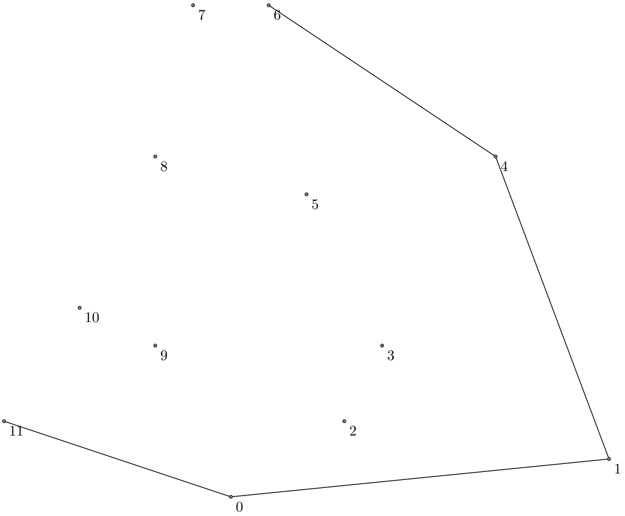
Kútpur hjúpur punktasafns



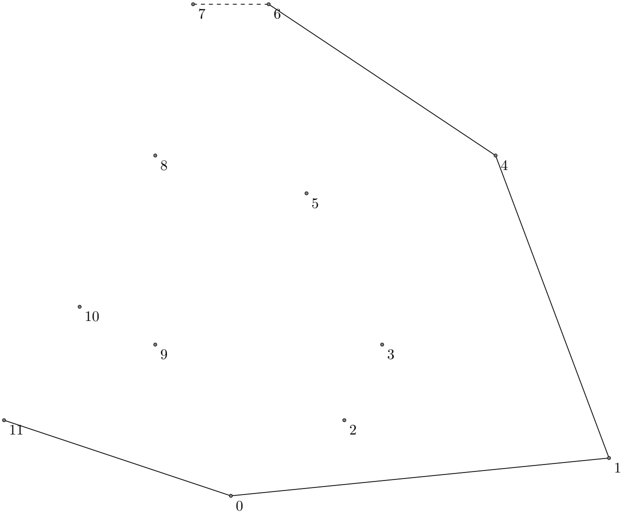
Kútpur hjúpur punktasafns



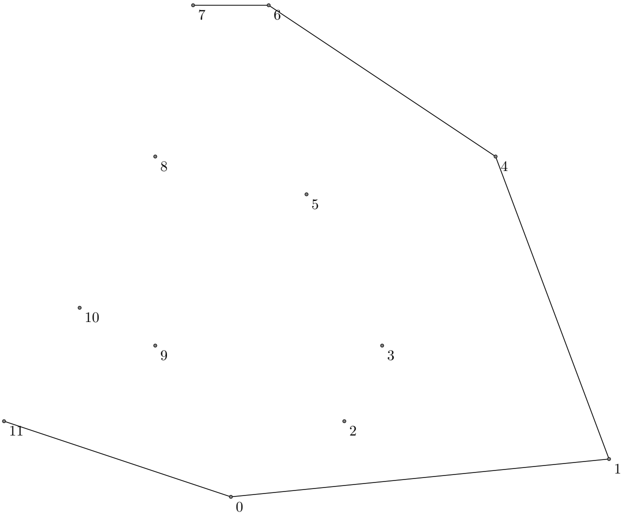
Kútpur hjúpur punktasafns



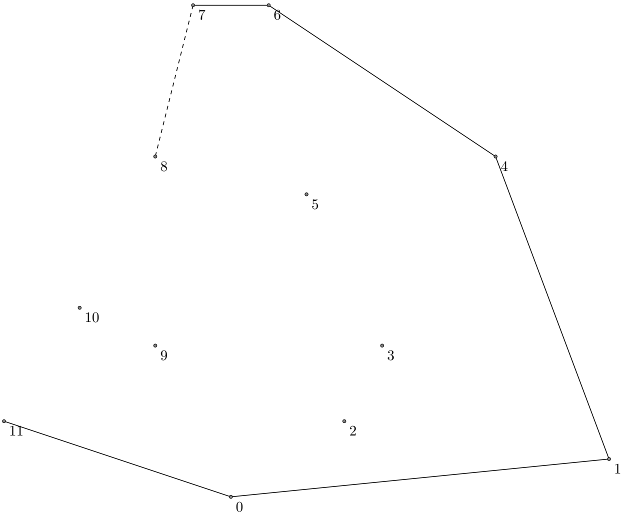
Kútpur hjúpur punktasafns



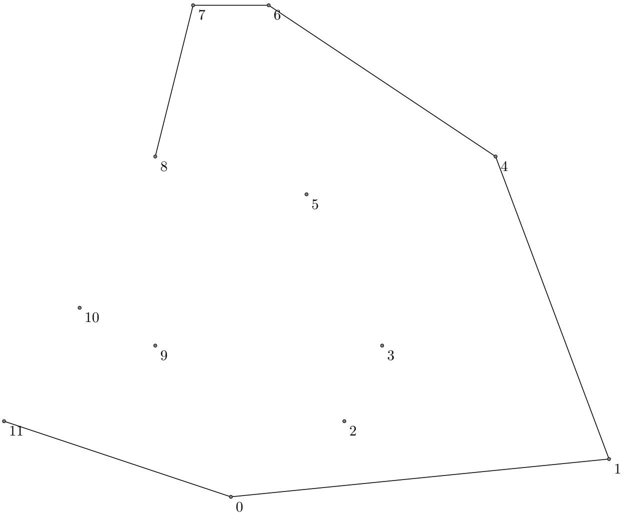
Kútpur hjúpur punktasafns



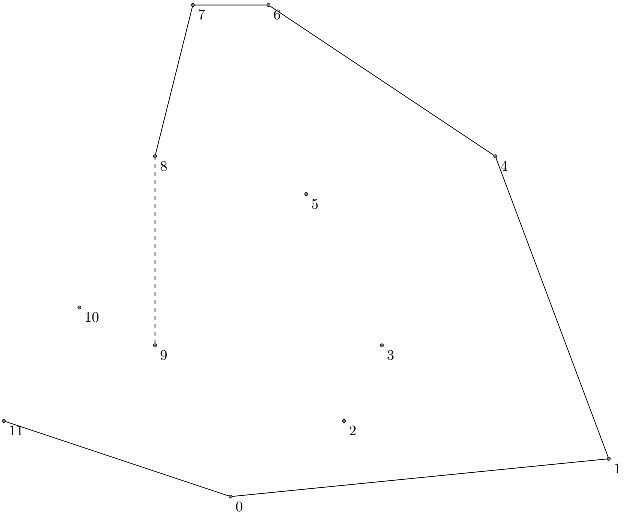
Kútpur hjúpur punktasafns



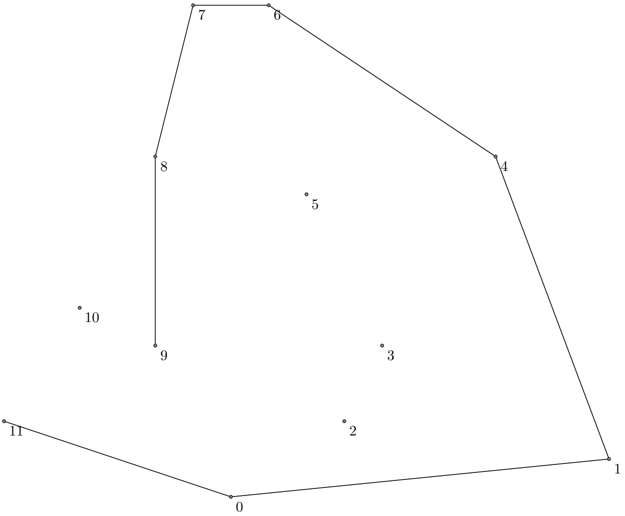
Kútpur hjúpur punktasafns



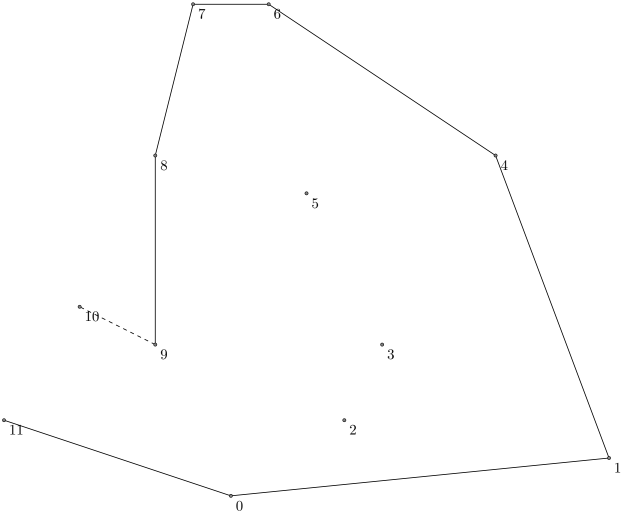
Kútpur hjúpur punktasafns



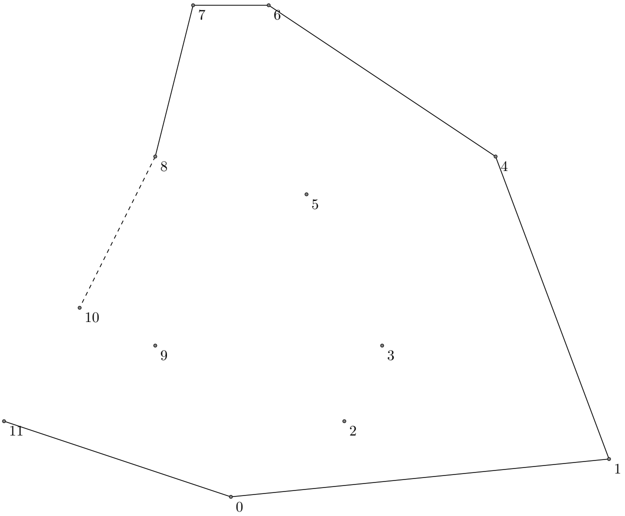
Kútpur hjúpur punktasafns



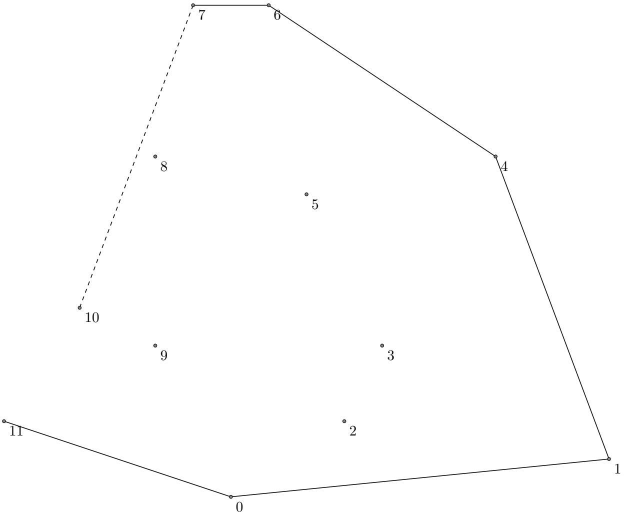
Kútpur hjúpur punktasafns



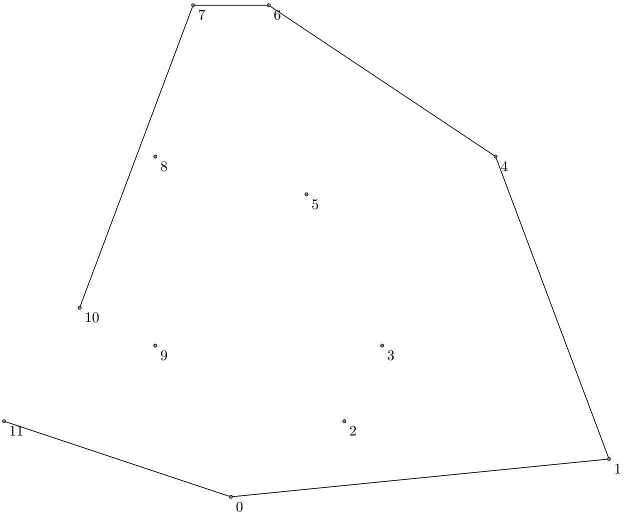
Kútpur hjúpur punktasafns



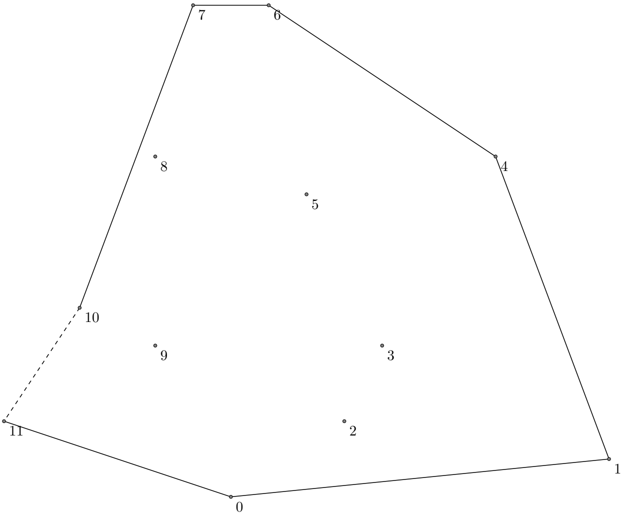
Kútpur hjúpur punktasafns



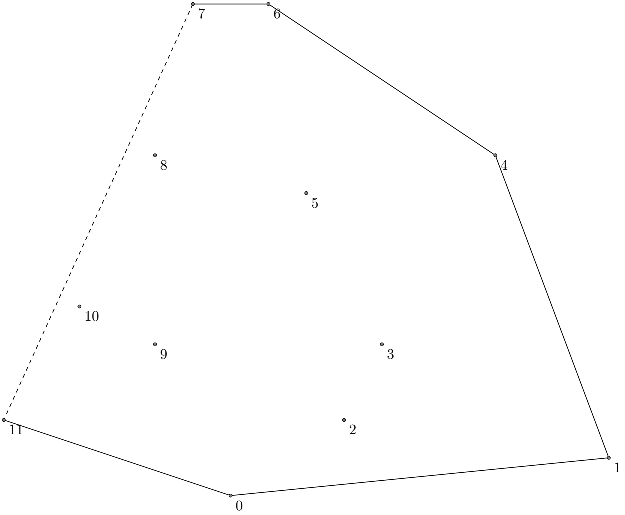
Kútpur hjúpur punktasafns



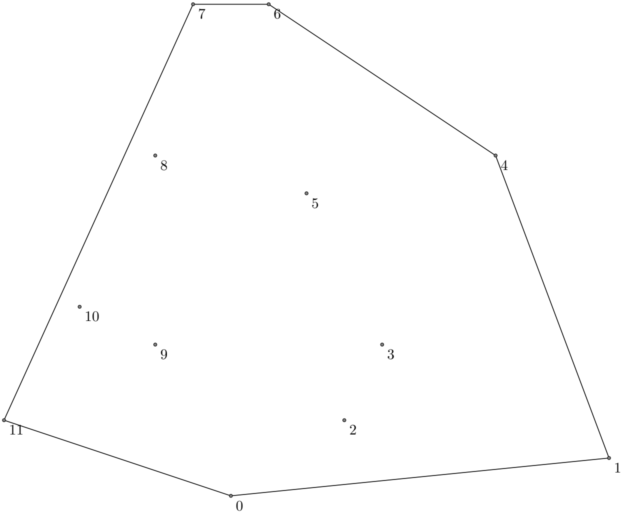
Kútpur hjúpur punktasafns



Kútpur hjúpur punktasafns



Kútpur hjúpur punktasafns



Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.

Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.
- Það er þó aðeins meira mál að sýna að þetta sé í raun kúpti hjúpur punktasafnsins.

Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.
- Það er þó aðeins meira mál að sýna að þetta sé í raun kúpti hjúpur punktasafnsins.
- Við látum það ógert í þessum fyrirlestri.

Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.
- Það er þó aðeins meira mál að sýna að þetta sé í raun kúpti hjúpur punktasafnsins.
- Við látum það ógert í þessum fyrirlestri.
- Ef punktasafnið inniheldur n punkta þá er reikniritið $\mathcal{O}(n \log n)$ út af því við þurfum að raða punktunum. Eftir röðun er reikniritið $\mathcal{O}(n)$ því hver punktur fer inn á hlaðan einu sinni.

Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.
- Það er þó aðeins meira mál að sýna að þetta sé í raun kúpti hjúpur punktasafnsins.
- Við látum það ógert í þessum fyrirlestri.
- Ef punktasafnið inniheldur n punkta þá er reikniritið $\mathcal{O}(n \log n)$ út af því við þurfum að raða punktunum. Eftir röðun er reikniritið $\mathcal{O}(n)$ því hver punktur fer inn á hlaðan einu sinni.
- **ATH:** Útfærslan á næstu glæru gerir ráð fyrir því að það sé til óúrkynjaður kúptur hjúpur, m.ö.o. til eru þrír punktar í safninu sem liggja ekki á sömu línu.

Graham's Scan

- Það er ljóst að hlaðinn í lok reikniritsins lýsir kúptum marghyrningi.
- Það er þó aðeins meira mál að sýna að þetta sé í raun kúpti hjúpur punktasafnsins.
- Við látum það ógert í þessum fyrirlestri.
- Ef punktasafnið inniheldur n punkta þá er reikniritið $\mathcal{O}(n \log n)$ út af því við þurfum að raða punktunum. Eftir röðun er reikniritið $\mathcal{O}(n)$ því hver punktur fer inn á hlaðan einu sinni.
- **ATH:** Útfærslan á næstu glæru gerir ráð fyrir því að það sé til óúrkynjaður kúptur hjúpur, m.ö.o. til eru þrír punktar í safninu sem liggja ekki á sömu línu.
- **Æfing:** Búið til fall sem ákvarðar hvort punkta safn hafi óúrkynjaðan kúpt hjúp (**Ábending:** minniháttar breytingar `ccw` gefa fall sem segir hvort gefnir punktar liggi á sömu línu).

Graham's Scan

```
point piv;
bool cmp(point a, point b)
{
    return fabs(arg(a - piv) - arg(b - piv)) < EPS ?
        abs(a - piv) < abs(b - piv) :
        arg(a - piv) < arg(b - piv);
}

int ccw(point a, point b, point c)
{
    return real(b - a)*imag(c - a) - imag(b - a)*real(c - a) > 0.0 ? 1 : 0;
}

polygon convex_hull(vector<point> p)
{
    polygon h; int j, i, mn = 0;
    for (i = 1; i < p.size(); i++)
        if (imag(p[i]) < imag(p[mn]) ||
                imag(p[i]) == imag(p[mn]) && real(p[i]) < imag(p[mn]))
            mn = i;
    swap(p[mn], p[0]); piv = p[0];
    sort(p.begin() + 1, p.end(), cmp);
    h.push_back(p[p.size() - 1]); h.push_back(p[0]); h.push_back(p[1]);
    i = 2;
    while (i < p.size())
    {
        j = h.size() - 1;
        if (ccw(h[j - 1], h[j], p[i])) h.push_back(p[i++]);
        else h.pop_back();
    }

    return h;
}
```

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.
- Sum ykkar hafa þó kannski ekki heyrt um þriðjungunarleit.

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.
- Sum ykkar hafa þó kannski ekki heyrt um þriðjungunarleit.
- Þriðjungunarleit finnur útgildi kúpts falls.

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.
- Sum ykkar hafa þó kannski ekki heyrt um þriðjungunarleit.
- Þriðjungunarleit finnur útgildi kúpts falls.
- Ólíkt helmingurnarleit þá skiptum við leitarsvæðinu upp í þriðjunga, í stað helminga, og notum til þess tvo punkta.

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.
- Sum ykkar hafa þó kannski ekki heyrt um þriðjungunarleit.
- Þriðjungunarleit finnur útgildi kúpts falls.
- Ólíkt helmingurnarleit þá skiptum við leitarsvæðinu upp í þriðjunga, í stað helminga, og notum til þess tvo punkta.
- Látum f vera kúpt fall á $[a, b]$.

Þriðjungunarleit

- Þið hafið öll heyrt um helmingunarleit.
- Sum ykkar hafa þó kannski ekki heyrt um þriðjungunarleit.
- Þriðjungunarleit finnur útgildi kúpts falls.
- Ólíkt helmingurnarleit þá skiptum við leitarsvæðinu upp í þriðjunga, í stað helminga, og notum til þess tvo punkta.
- Látum f vera kúpt fall á $[a, b]$.
- Þá gildir fyrir $s, t \in [a, b]$, $s < t$ að ef $f(s) > f(t)$ þá tekur f lágildi á $[s, b]$, en annars tekur f lágildi á $[a, t]$.

Þriðjungunar leit

```
// lagmorkum fall f a bili [a, b]
double l = a, r = b;
while (r - l > EPS)
{
    m1 = l + (r - l)/3.0;
    m2 = r - (r - l)/3.0;
    if (f(m1) > f(m2)) l = k1;
    else r = k2;
}
// nu er f(r) lagildi f á [a, b]
```

Niðurlag

- Við erum nú búnir að fara yfir það efni sem við ætluðum okkur.

Niðurlag

- Við erum nú búnir að fara yfir það efni sem við ætluðum okkur.
- Ef ykkur vantar námskeið fyrir komandi annir sem tengjast lauslega keppnisforritun eru t.d. í boði *Þýðendur*, *Líkinda og tölfræði*, *Fléttufræði*, *Rúmfræði*, *Algebra II*, *Slembiferli*, *Tvinnfallagreining (Stærðfræðigreining IIIB)* og *Töluleg greining*.

Niðurlag

- Við erum nú búnir að fara yfir það efni sem við ætluðum okkur.
- Ef ykkur vantar námskeið fyrir komandi annir sem tengjast lauslega keppnisforritun eru t.d. í boði *Þýðendur*, *Líkinda og tölfræði*, *Fléttufræði*, *Rúmfræði*, *Algebra II*, *Slembiferli*, *Tvinnfallagreining (Stærðfræðigreining IIIB)* og *Töluleg greining*.
- Við mælum sérstaklega með *Algebra I*, *Greining reiknirita* og *Netafræði*.