# Geometriai algoritmusok

*(Horváth Gyula és Szlávi Péter előadásai felhasználásával)*

# Geometriai algoritmusok

## Alapfogalmak

**Pont**: $(x;y) \in RxR$

Megjegyzés: Csak olyan feladatokat tekintünk, ahol a pontok koordinátái mindig egész számok, és a megoldásához nem kell lebegőpontos aritmetikát használni.

**Szakasz**

A $p_1$ és $p_2$ pont által meghatározott szakasz:

$$\{x = a*p_1.x+(1-a)*p_2.x;\ y = a*p_1.y+(1-a)*p_2.y)\}$$

# Geometriai algoritmusok

## Egyenes

1. y = mx+b egyenlettel: azon (x;y) pontok halmaza, amelyekre teljesül az egyenlet.
2. ax+by+c = 0 egyenlettel.
3. Egyenes megadása két pontjával: e(p1; p2)

Pontok ábrázolására a

```
Pont=Rekord (x,y: egész; az: egész)
```

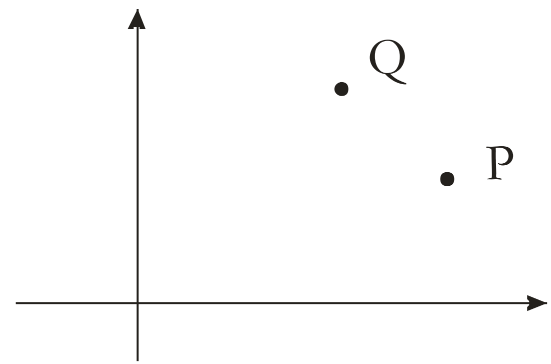
típust használjuk, ahol p.az a pont azonosítója, vagy sorszáma a feladatleírás szerint (ha szükség van rá).

# Geometriai algoritmusok

## Feladat:

Adjuk meg, hogy az origóból nézve az 1. sík-negyedbe eső P ponthoz képest a Q balra, jobbra vagy pedig egy irányban látszik-e!

$$\text{Irány(P,Q)} = \begin{cases} -1, & \text{ha} \quad \text{balra} \\ +1, & \text{ha} \quad \text{jobbra} \\ \phantom{+}0, & \text{ha} \quad \text{egy irányban} \end{cases}$$



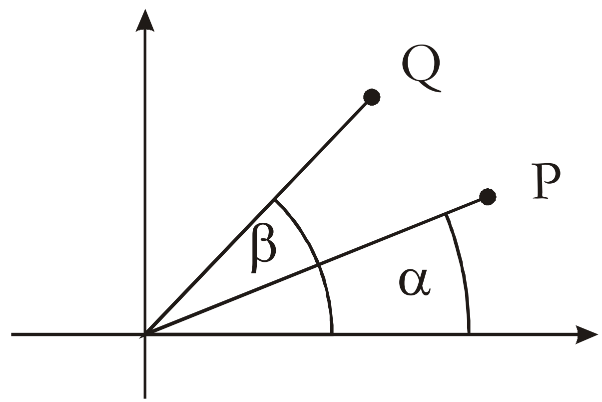

Ponttípus:

Típus Tpont=rekord(x,y: Egész)

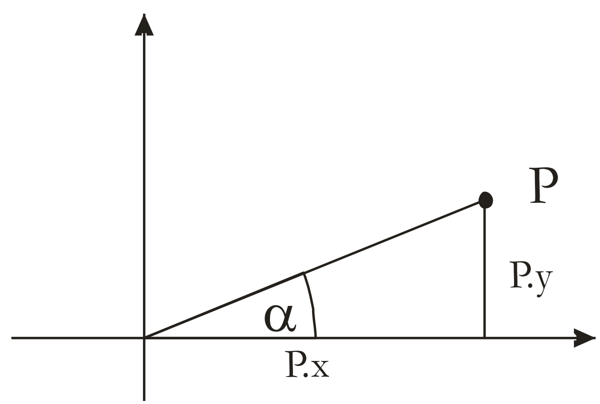
# Értelmezés:

A pontok irányát megadhatjuk az oda vezető egyenes és az x-tengely szögével.

$\alpha<\beta \rightarrow \tan(\alpha)<\tan(\beta)$




$\tan(\alpha)=P.y/P.x$




# Geometriai algoritmusok

$$\alpha<\beta \leftrightarrow \tan(\alpha)<\tan(\beta) \leftrightarrow P.y/P.x<Q.y/Q.x \leftrightarrow$$
$$P.y*Q.x<Q.y*P.x \leftrightarrow P.y*Q.x-Q.y*P.x<0$$

**Állítás:**

$$\text{Irány}(P,Q)=\text{sgn}(P.y*Q.x-Q.y*P.x)$$

(és ez igaz nem csak az 1. síknegyedben!).

$$\text{sgn}(P.y*Q.x-Q.y*P.x)=\begin{cases}-1, & \text{ha Q a P-től balra}\\ +1, & \text{ha Q a P-től jobbra}\\ 0, & \text{ha Q és P egy irányban}\end{cases}$$

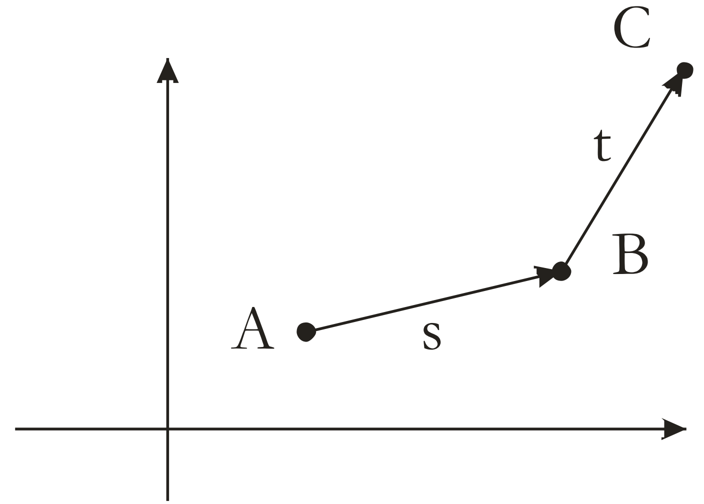
# Geometriai algoritmusok

```
Irány(P,Q):
   S:=P.y*Q.x-Q.y*P.x
   Ha S<0 akkor Irány:=-1
   különben ha S=0 akkor Irány:=0
   különben Irány:=1
Függvény vége.
```

# Geometriai algoritmusok

## Feladat:

Egy **s** (A→B) szakaszhoz képest egy **t** (B→C) szakasz milyen irányban fordul?




## Megoldásötlet:

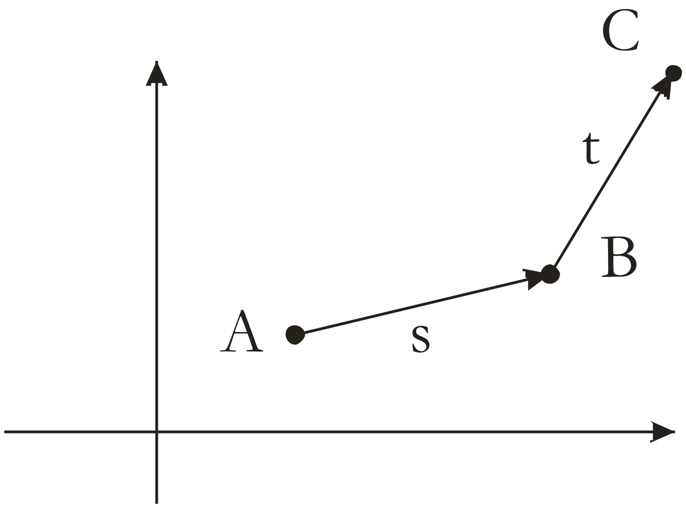
Toljuk el az **s**-t és a **t**-t úgy, hogy az A pont az origóba kerüljön! Ezzel visszavezetjük az „irányos” feladatra!

Fordul(A,B,C)=Irány(B–A,C–A)

Ezzel ekvivalens feladat: *Az (A,B)-n átmenő egyenestől a C pont balra van, vagy jobbra van, vagy az (A,B) egyenesen van?*

# Geometriai algoritmusok

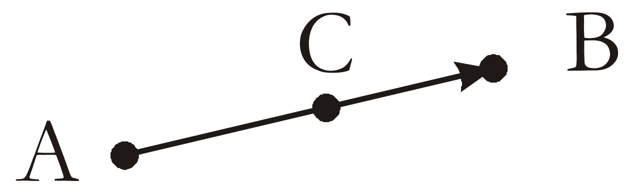
```
Fordul(A,B,C):
  P:=B-A          {P.x:=B.x-A.x; P.y:=B.y-A.y}
  Q:=C-A          {Q.x:=C.x-A.x; Q.y:=C.y-A.y}
  Fordul:=Irány(P,Q)
Függvény vége.
```




# Geometriai algoritmusok

## Feladat:

Döntsük el, hogy egy C pont rajta van-e egy (A,B) szakaszon!

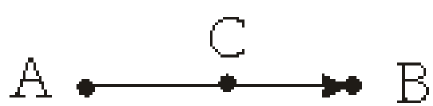


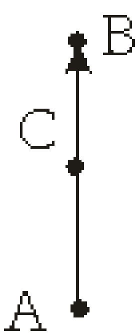

## Megoldás:

- Biztos nincs rajta, ha az A-B-C úton valamerre fordulni kell!
- Ha nem kell fordulni, akkor A és B között kell lennie!

# Geometriai algoritmusok

```
Rajta(a,b,c):
  Rajta:=Fordul(a,b,c)=0 és Közte(a.x,c.x,b.x)
         és Közte(a.y,c.y,b.y)
Függvény vége.
Közte(r,s,t):
  Közte:=r≤s és s≤t vagy t≤s és s≤r
Függvény vége.
```

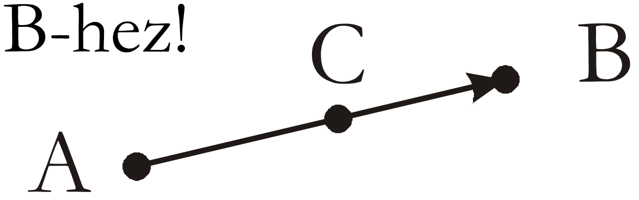



# Geometriai algoritmusok

**Feladat:**

Döntsük el, hogy egy az (A,B) szakaszon levő C pont közelebb van-e A-hoz, mint B-hez!




**Megoldás:**

C közelebb van A-hoz, ha

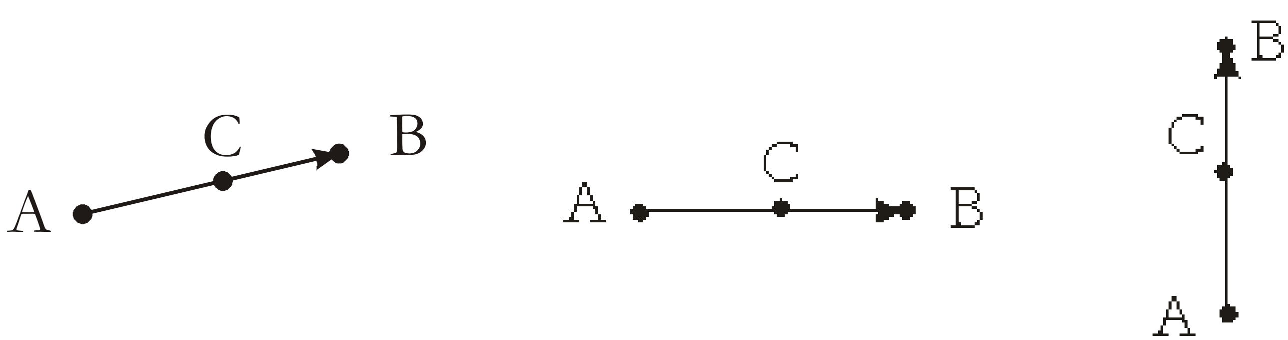
- x-koordináta szerint közelebb van;

vagy

- y-koordináta szerint közelebb van.

# Geometriai algoritmusok

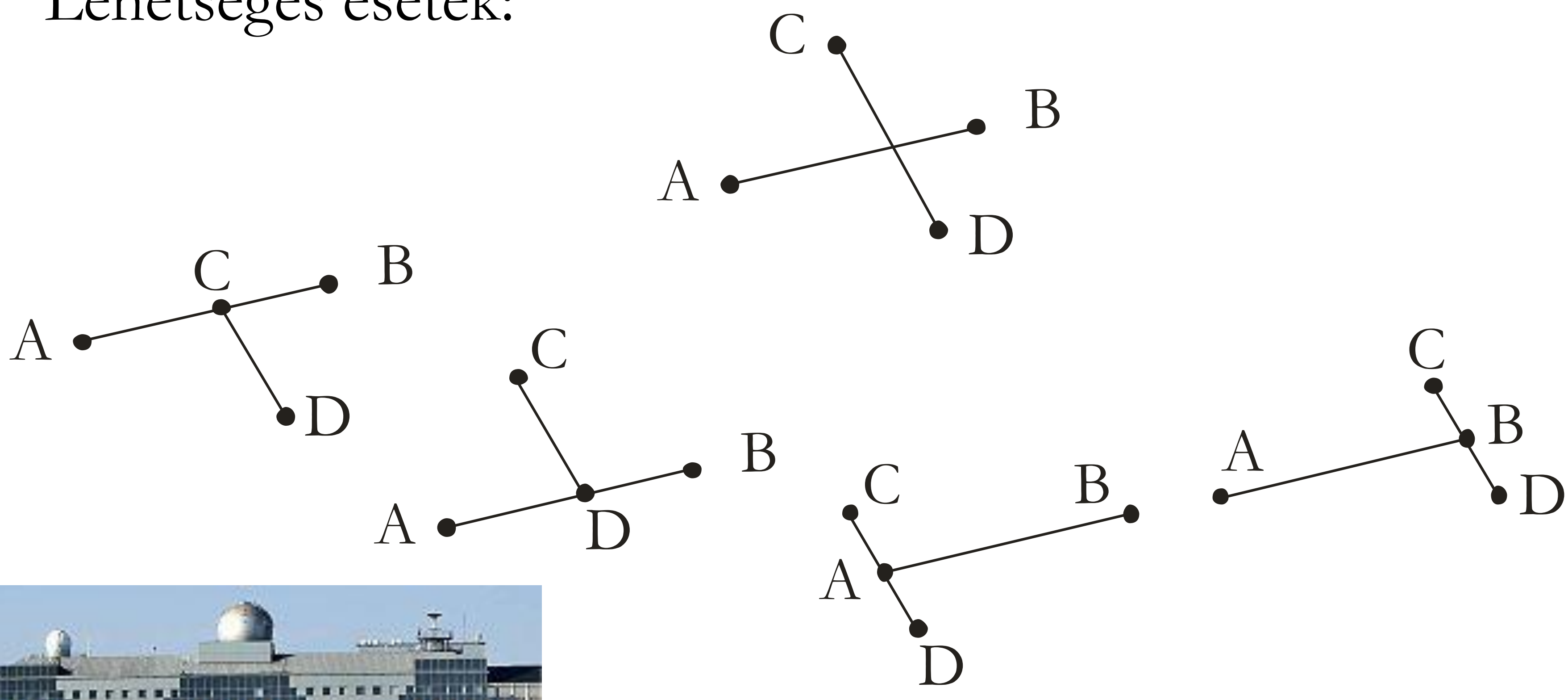
```
Közelebb(a,b,c):
  Közelebb:=|a.x-c.x|<|b.x-c.x| vagy
            |a.y-c.y|<|b.y-c.y|
Függvény vége.
```



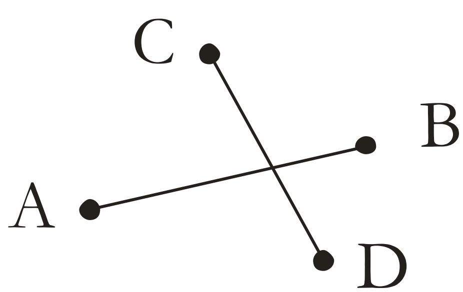

## Feladat:

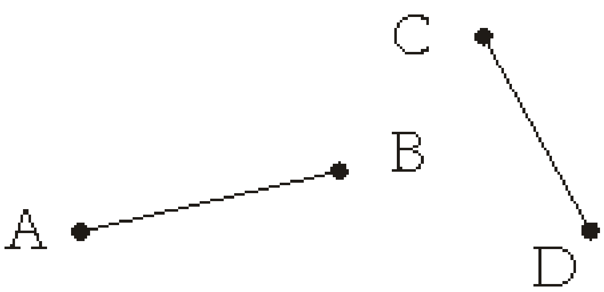
Döntsük el, hogy az (A,B) szakasz metszi-e a (C,D) szakaszt!

Lehetséges esetek:




```
Metszi(A,B,C,D):
  Metszi:=Fordul(A,B,C)*Fordul(A,B,D)<0 és
          Fordul(C,D,A)*Fordul(C,D,B)<0 vagy
          Rajta(A,B,C) vagy Rajta(A,B,D) vagy
          Rajta(C,D,A) vagy Rajta(C,D,B)
Függvény vége.
```



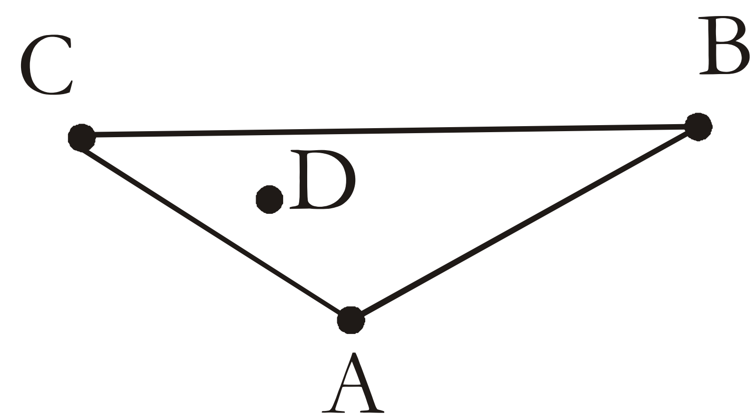




## Feladat:

Döntsük el, hogy a D pont az (A,B,C) háromszög belsejében van-e!

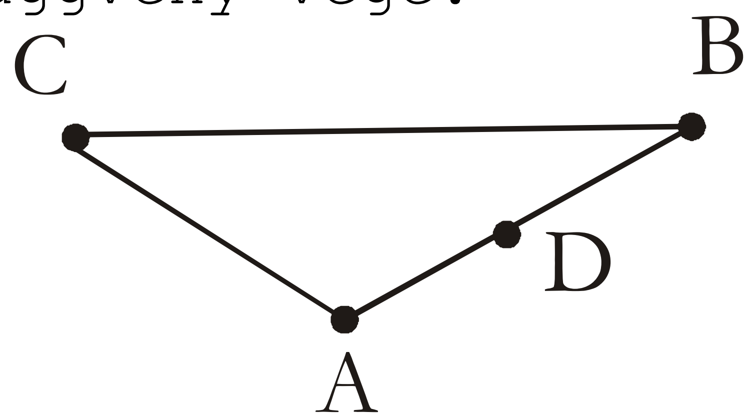
## Megoldásötlet:

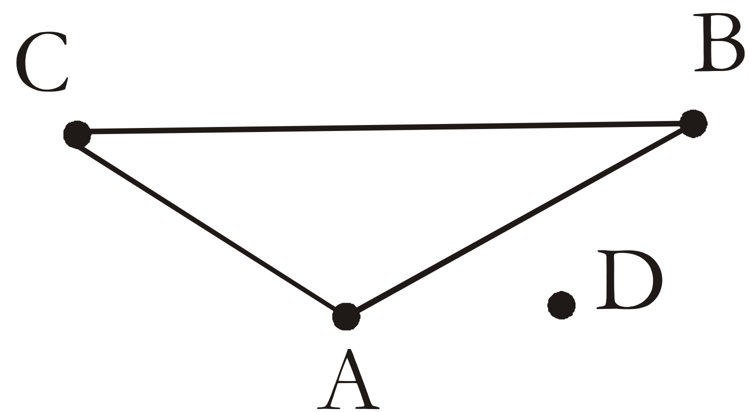
Belül van, ha a háromszöget A→B→C→A sorrendben körbejárva a D pont vagy mindig balra, vagy mindig jobbra van.

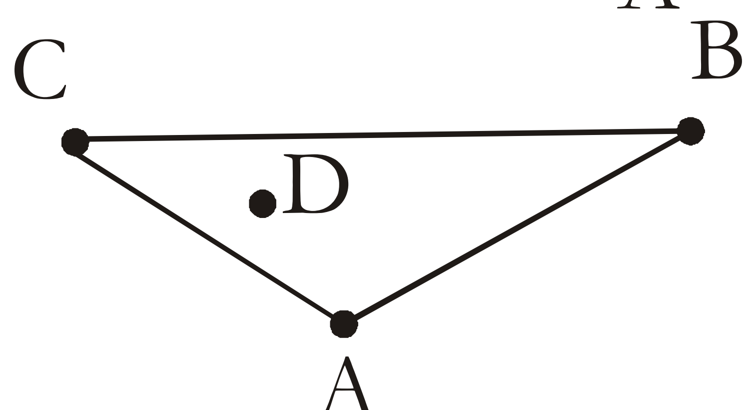



# Geometriai algoritmusok

```
Belül(A,B,C,D):
   Belül:=Fordul(A,B,D)=Fordul(B,C,D)
             és Fordul(B,C,D)=Fordul(C,A,D)
Függvény vége.
```



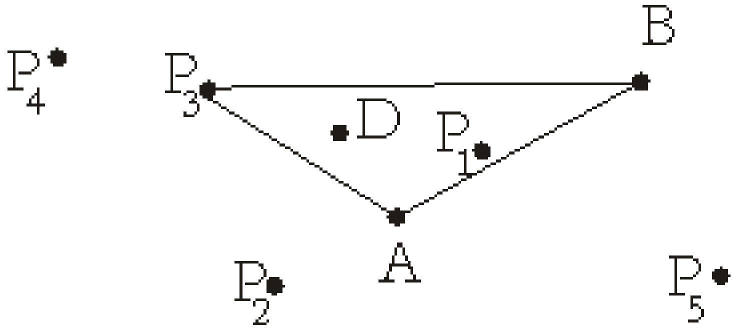




## Feladat:

Adott A, B és D pont esetén adjunk meg további N pont közül egy $P_i$ pontot úgy, hogy a D pont az $(A,B,P_i)$ háromszög belsejében legyen!

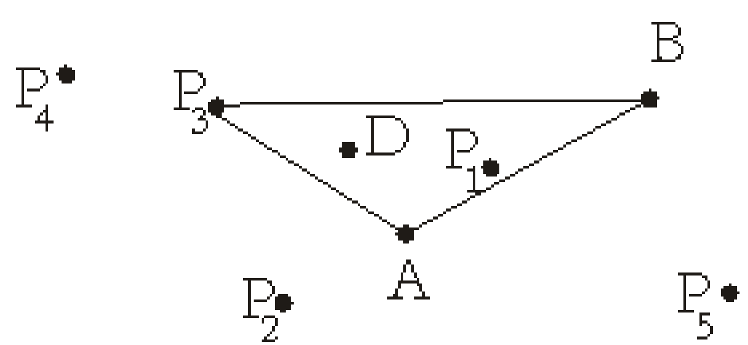
## Megoldásötlet:

Belül van a $P_i$ pont, ha a háromszöget $A \rightarrow B \rightarrow P_i \rightarrow A$ sorrendben körbejárva a D pont vagy mindig balra, vagy mindig jobbra van.




# Geometriai algoritmusok

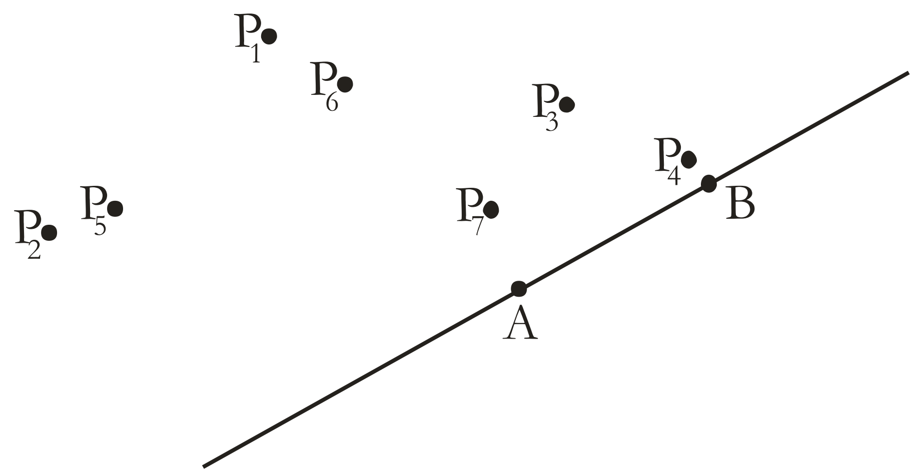
```
Keresés(A,B,N,P,D,Van,S):
  ir:=Fordul(A,B,D); S:=1
  Ciklus amíg S≤N és (Fordul(B,P(i),D)≠ir vagy
                       Fordul(P(i),A,D)≠ir)
    S:=S+1
  Ciklus vége
  Van:=S≤N
Eljárás vége.
```




## Feladat:

Adott A és B pont esetén adjunk meg további N pont közül egy $P_i$ pontot úgy, hogy az $(A,B,P_i)$ háromszög belsejében egyetlen más pont se legyen!

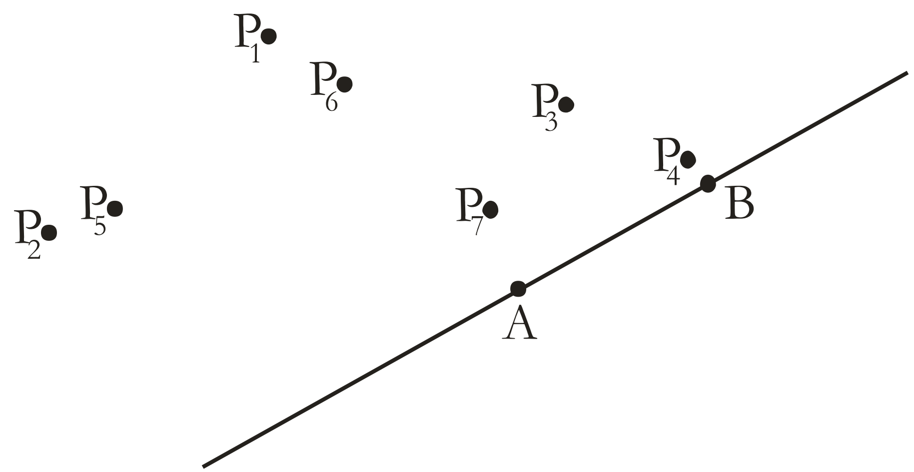
Feltehető, hogy az összes pont az (A,B) egyenestől balra van!




# Geometriai algoritmusok

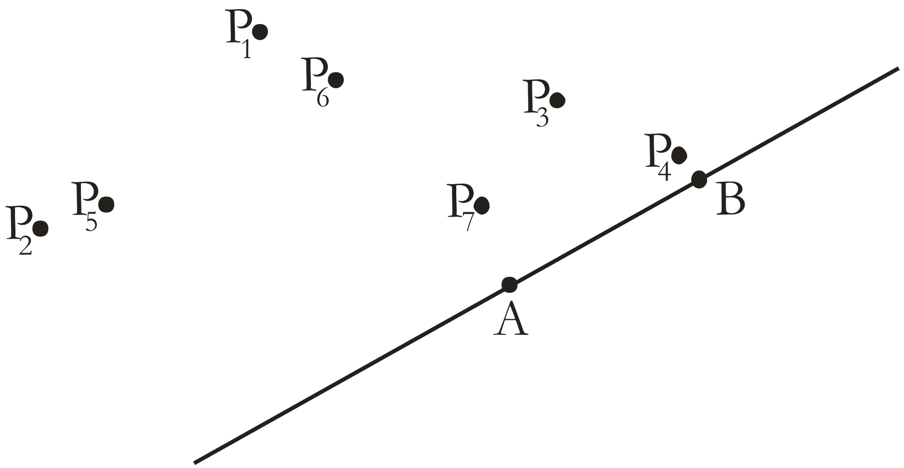
## Megoldás:

Ha van egy potenciális jelöltünk (pl. $P_1$), akkor az $(A,P_1)$-től balra levők és a $(B,P_1)$-től jobbra levők biztos nincsenek az $(A,B,P_1)$ háromszögben!




# Geometriai algoritmusok

```
Kiválasztás(A,B,N,P,D,S):
  S:=1
  Ciklus i=2-től N-ig
    Ha Fordul(A,P(S),P(i))=1 és
       Fordul(B,P(S),P(i))=-1
     akkor S:=i
  Ciklus vége
Eljárás vége.
```

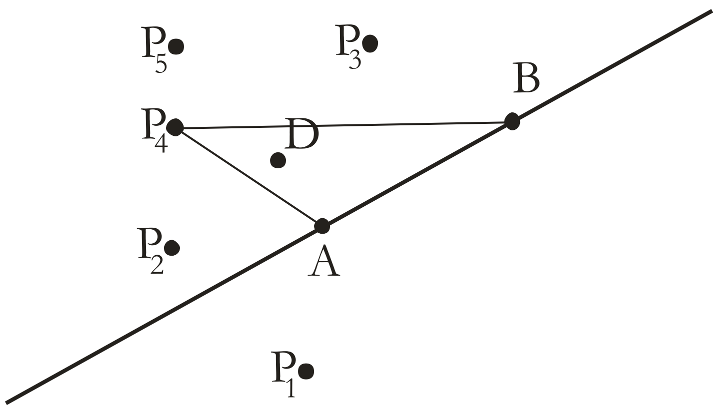



# Geometriai algoritmusok

## Feladat:

Adott A, B és D pont esetén adjunk meg további N pont közül egy $P_i$ pontot úgy, hogy a D pont az ($A,B,P_i$) három-szög belsejében legyen, de egyetlen más pont se legyen benne!

Feltehető, hogy a D pont az (A,B) egyenestől balra van!




## Megoldás:

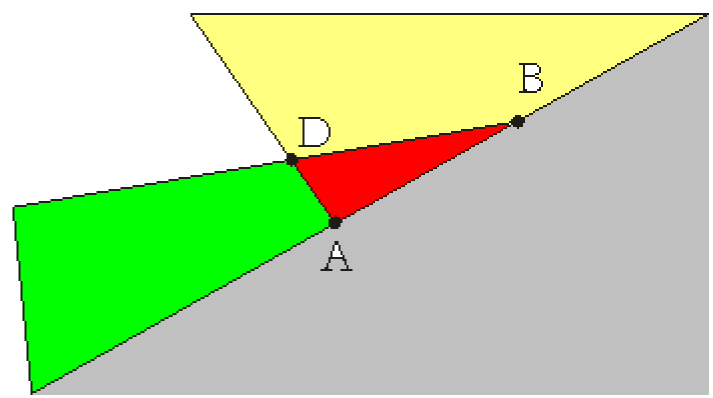
A feladat nem oldható meg, ha a piros színű tartományban van pont – ez a beolvasásnál kiszűrendő.

Az (A,B)-től jobbra levő pontok nem jók – szürke tartomány.

Az (A,D)-től jobbra levő pontok nem jók – sárga tartomány.

A (B,D)-től balra levő pontok nem jók – zöld tartomány.

Sajnos nem a teljes fehér tartomány jó!



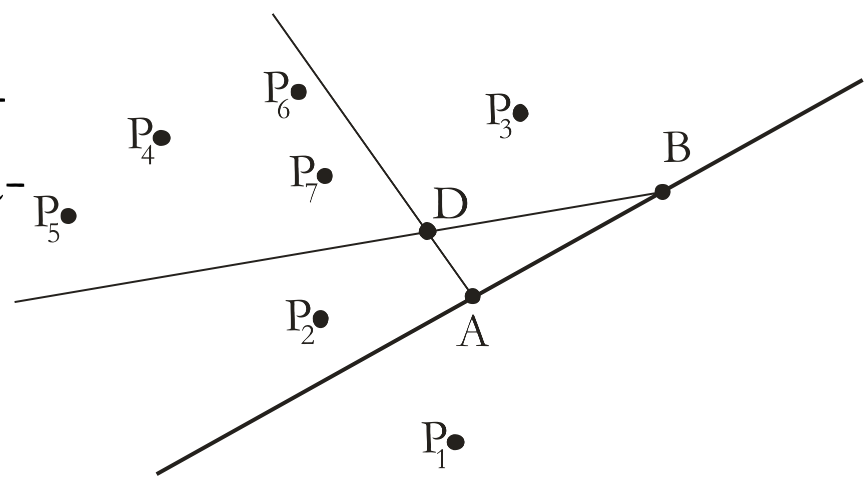

# Geometriai algoritmusok

## Megoldás:

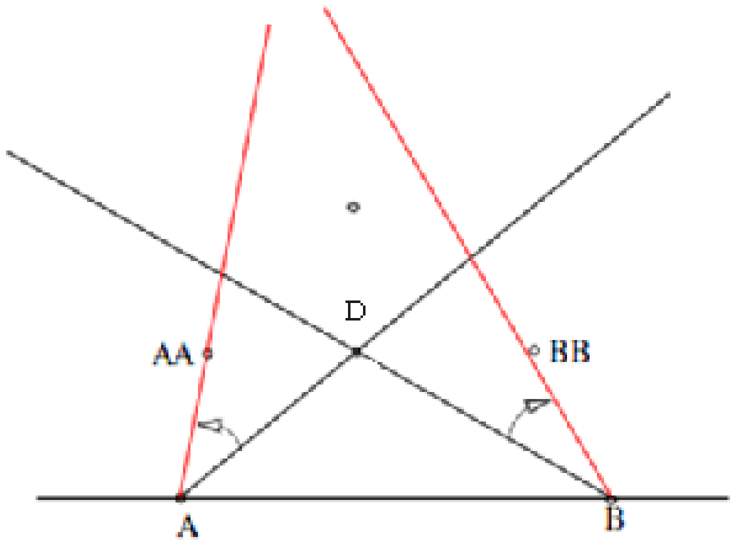
Az (A,D) egyenestől balra, a (B,D) egyenestől jobbra olyan $P_i$ pontot kell találnunk, hogy az $(A,B,P_i)$ háromszög belsejében ne legyen más pont!

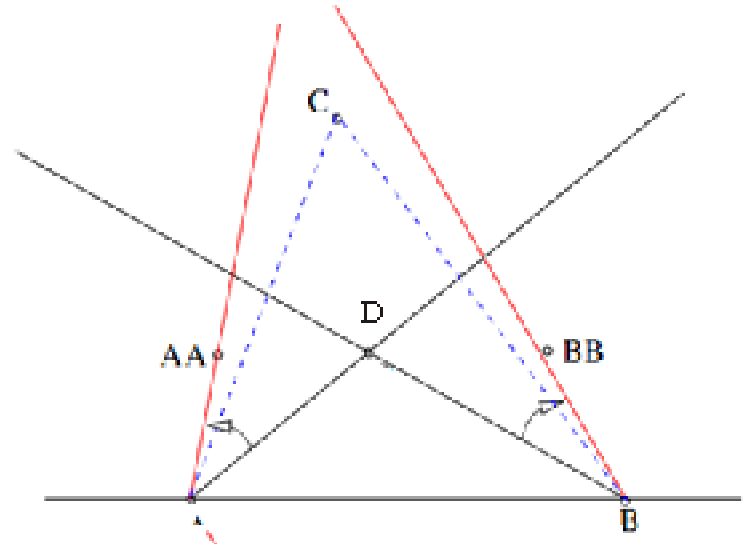
Ha van egy potenciális jelöltünk (pl. $P_4$), akkor az $(A,P_4)$-től balra levők és a $(B,P_4)$-től jobbra levők biztos nincsenek az $(A, B,P_4)$ háromszögben!

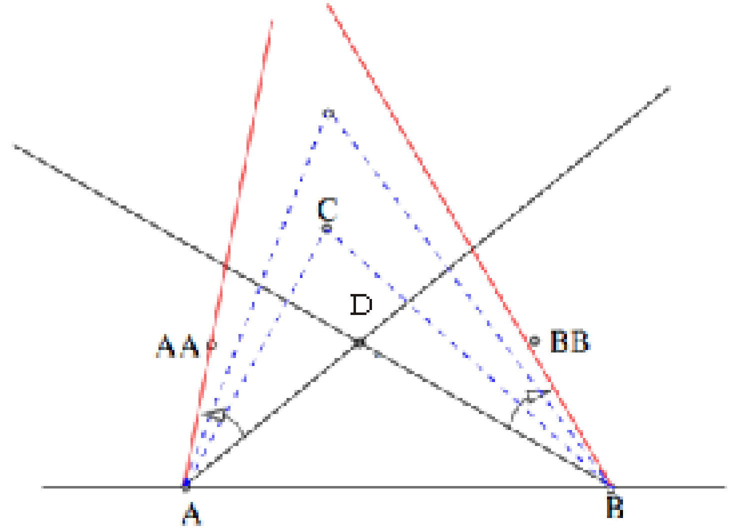
Most $P_5$ jó lenne, de ha $P_2$-t közelítenénk D-hez, akkor $P_2$ a háromszög belsejébe kerülne.




# Geometriai algoritmusok



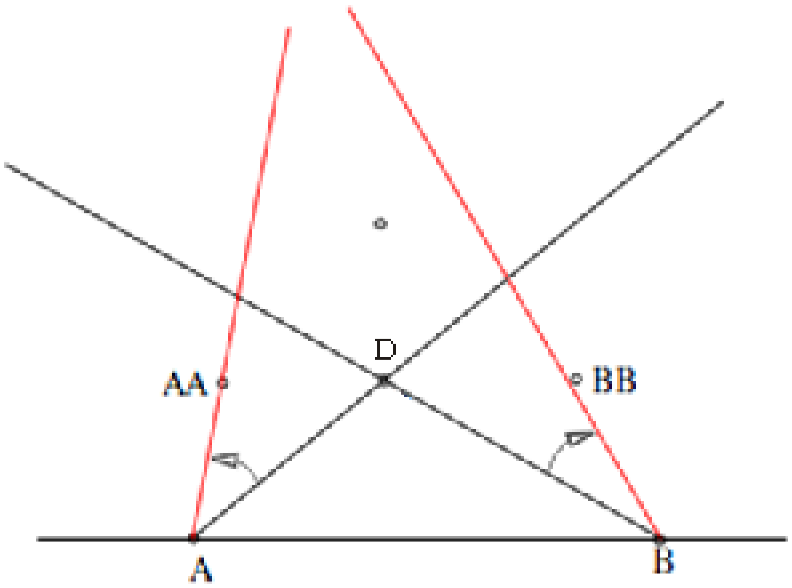




# Geometriai algoritmusok

```
Keresés(A,B,N,P,D,Van,S):
  Ha Fordul(A,B,D)>0 akkor Csere(A,B)
  AA:=0; BB:=0; C:=0
  Ciklus i=1-től N-ig {AA és BB meghatározása}
    FAB:=Fordul(A,B,P(i))
    FBD:=Fordul(B,D,P(i))
    FDA:=Fordul(D,A,P(i))
    Ha FAB≤0 és FBD≤0 és FDA≤0 akkor ...
{nincs, ezt már a beolvasásnál kiszűrhetnénk}
```

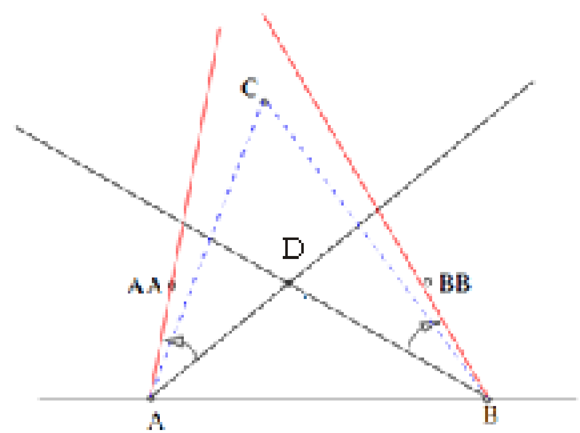
# Geometriai algoritmusok

```
    Ha FAB>0 és FDA<0 és FBD>0 akkor
       Ha AA=0 vagy Fordul(A,P(AA),P(i))<0
          akkor AA:=i
    Ha FAB>0 és FDA>0 és FBD<0 akkor
       Ha BB=0 vagy Fordul(B,P(BB),P(i))>0
          akkor BB:=i
Ciklus vége
…
```



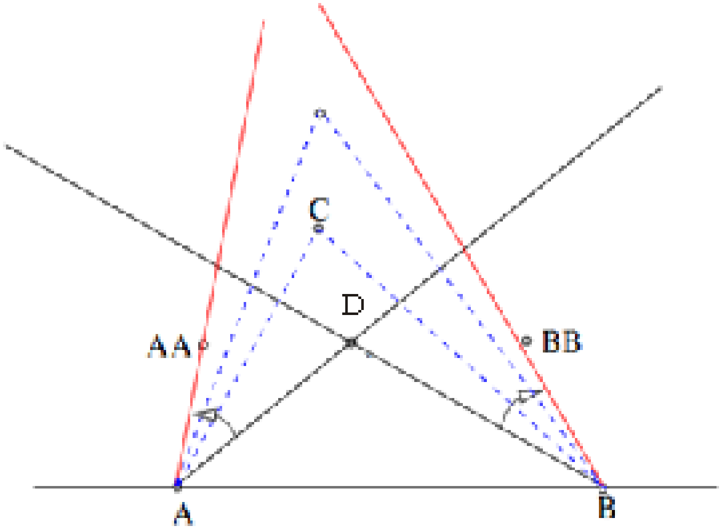

# Geometriai algoritmusok

```
i:=1; C:=0
Ciklus amíg i≤n és C≠0 {egy C meghatározása}
   Ha i≠AA és i≠BB akkor
      Ha AA>0 akkor FAAA:=Fordul(A,P(AA),P(i))
      Ha BB>0 akkor FBBB:=Fordul(B,P(BB),P(i))
      FDA:=Fordul(D,A,P(i))
      FBD:=Fordul(B,D,P(i))
      Ha (AA=0 vagy FAAA<0) és (BB=0 vagy FBBB>0)
          és FDA<0 és FBD<0 akkor C:=i;
Ciklus vége
Ha C>0 akkor … {C finomítása}
```




# Geometriai algoritmusok

```
  … {C finomítása}
  Ciklus i=C+1-től n-ig
    Ha i≠AA és i≠BB akkor
       FAB:=Fordul(A,B,P(i))
       FAC:=Fordul(A,P(C),P(i))
       FBC:=Fordul(B,P(C),P(i))
       Ha FAB>0 és FAC<0 és FBC>0 akkor C:=i
  Ciklus vége
Eljárás vége.
```




# Geometriai algoritmusok

## Feladat:

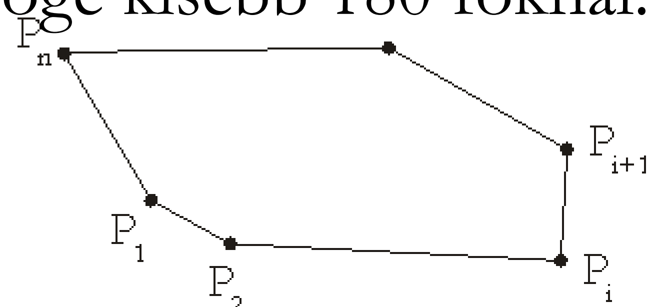
Döntsük el, hogy a $(P_1,\ldots,P_n)$ sokszög konvex sokszög-e!

## Definíció:

A sokszög **konvex**, ha bármely egyenes, mely a sokszögön áthalad, (és nem érinti egy élben vagy csúcsban) pontosan kétszer metszi azt.

A sokszög **konvex**, ha minden olyan szakasz, ami a sokszög két belső pontját köti össze, a sokszög belsejében halad.

A sokszög **konvex**, ha minden szöge kisebb 180 foknál.



# Geometriai algoritmusok

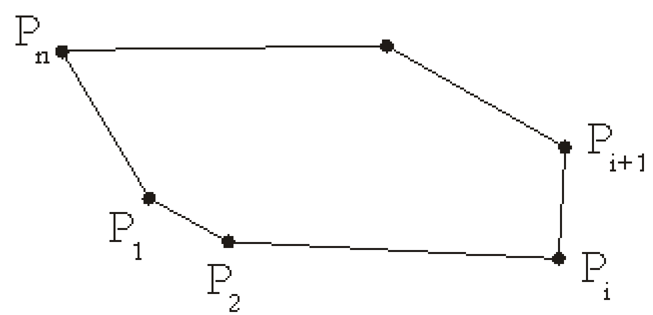
## Feladat:

Döntsük el, hogy a $(P_1,\ldots,P_n)$ sokszög konvex sokszög-e!

## Megoldás:

Az első 2 definíció nem ad hatékony megoldást.

A hatékony megoldás a harmadik definíció alapján: Az óramutató járásával ellentétes körbejárással haladva minden csúcsban balra kell fordulni!




# Geometriai algoritmusok

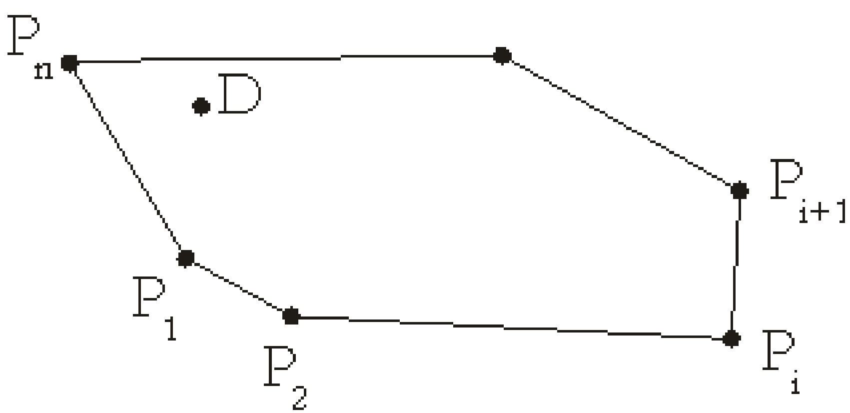
```
Konvex(P,N)
  P(N+1):=P(1); P(N+2):=P(2); i:=1
  Ciklus amíg i≤N és Fordul(P(i),P(i+1),P(i+2))<0
    i:=i+1
  Ciklus vége
  Konvex:=i>N
Eljárás vége.
```

## Feladat:

Döntsük el, hogy a D pont a $(P_1,\ldots,P_n)$ konvex sokszög belsejében van-e!

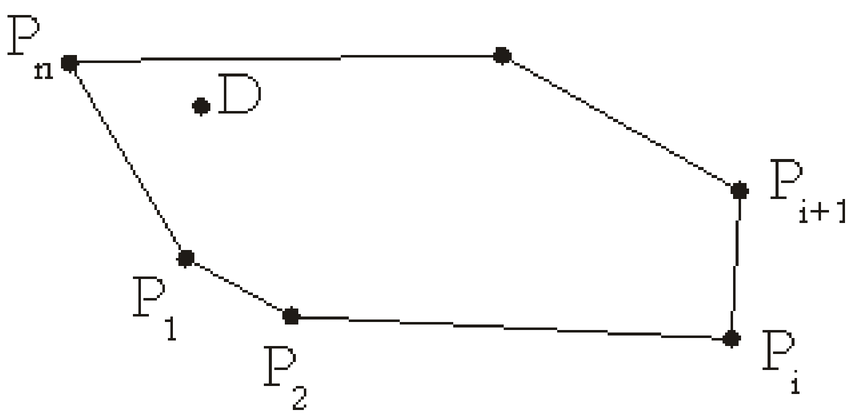
## Megoldásötlet:

Belül van, ha a sokszöget adott sorrendben körbejárva a D pont vagy mindig balra, vagy mindig jobbra van.



# Geometriai algoritmusok

```
Belül(N,P,D):
  P(N+1):=P(1); Ir:=Fordul(P(1),P(2),D); i:=2
  Ciklus amíg i≤N és Ir=Fordul(P(i),P(i+1),D)
    i:=i+1
  Ciklus vége
  Belül:=i>N
Függvény vége.
```



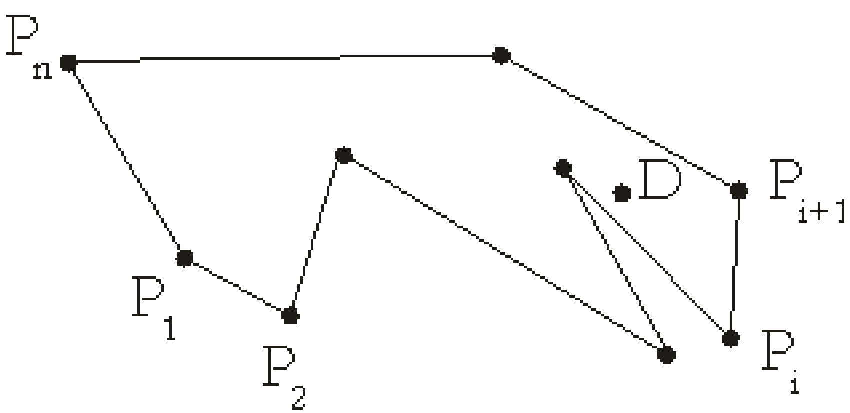

# Geometriai algoritmusok

## Feladat:

Döntsük el, hogy a D pont a ($P_1,\ldots,P_n$) tetszőleges sokszög belsejében van-e!

## Probléma:

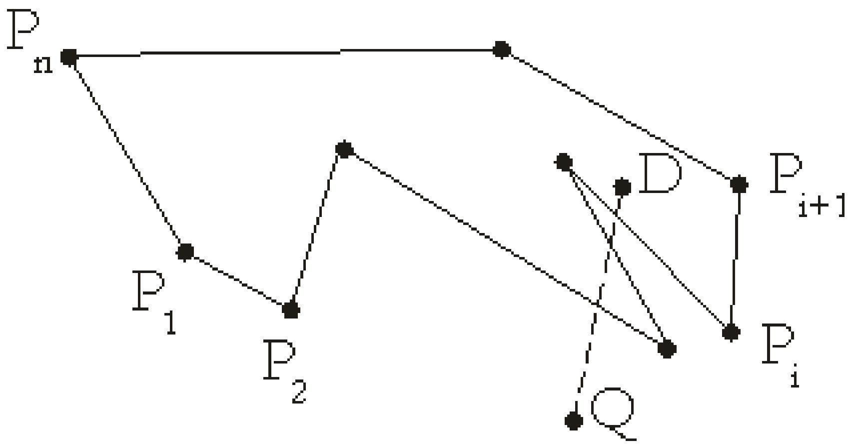
Itt nem működik a konvex esetben alkalmazható: mindig egy irányban van elv.




# Geometriai algoritmusok

## Megoldás:

Kössük össze a D pontot egy biztosan külső Q ponttal, majd számoljuk meg, hogy a (D,Q) szakasz a sokszög hány oldalát metszi!




$$Q.y := \min_{i=1,\dots,N}\left(P_i.y\right) - 1$$

$$Q.x := \max_{\substack{i=1,\dots,N \\ P_i.x<D.x}}\left(P_i.x\right)$$

# Geometriai algoritmusok

```
Külső(N,P,D,Q):
  Q.y:=P(1).y; Q.x:=-∞
  Ciklus i=2-től N-ig
    Ha Q.y>P(i).y akkor Q.y:=P(i).y
    Ha P(i).x<D.x akkor
       Ha Q.x<P(i).x akkor Q.x:=P(i).x
  Ciklus vége
  Q.y:=Q.y-1
Függvény vége.
```

Ha Q.x=-∞ maradt, akkor a D pont kívül van!

# Geometriai algoritmusok

```
Belül(N,P,D):
  P(N+1):=P(1); Db:=0
  Külső(N,P,D,Q)
  Ciklus i=1-től N-ig
    Ha Metszi(P(i),P(i+1),D,Q) akkor Db:=Db+1
  Ciklus vége
  Belül:=(Db mod 2)=1
Függvény vége.
```

# Geometriai algoritmusok

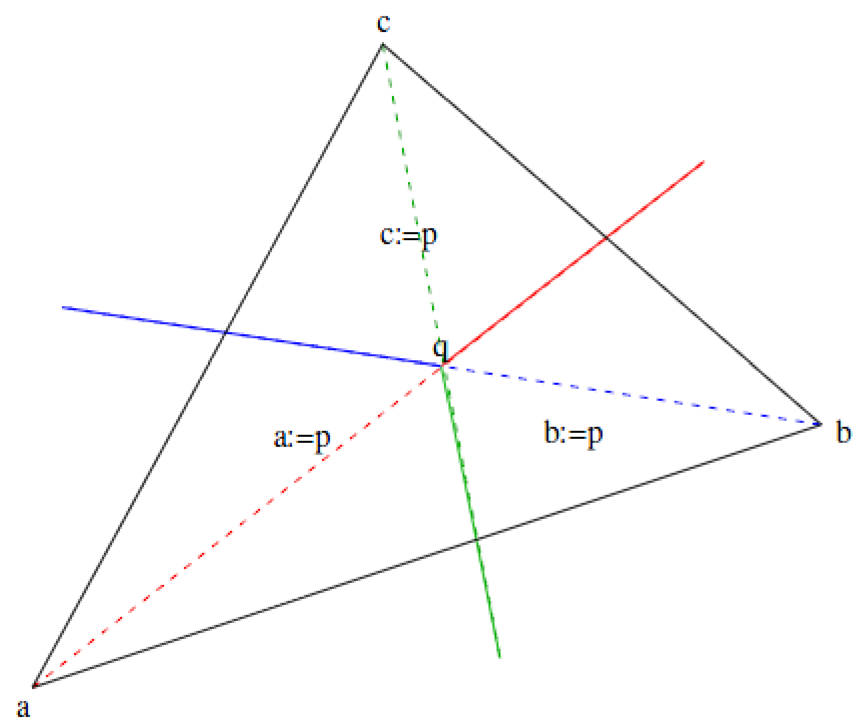
## Feladat:

Adott Q pont esetén adjunk meg további N pont közül három olyan A,B,C pontot úgy, hogy az (A,B,C) háromszög tartalmazza a Q pontot, de egyetlen más pontot sem! A határvonala is legyen része a háromszögnek!

1. Állítás. Ha van olyan (tetszöleges) A, B, C pont, hogy Q az (A,B,C) háromszögbe, vagy oldalára esik, akkor ez finomítható úgy, hogy a feltétel teljesüljön.
2. Állítás. Akkor és csak akkor van megoldás, ha a Q pont a ponthalmaz konvex burkán belül vagy az oldalán van.

## Megoldás:

A háromszög finomítása. Minden p pontra, amely az (A,B,C) háromszögbe, vagy oldalára esik, Q∈(A,B,P), vagy Q∈(B,C,P) vagy Q∈(C,A,P) teljesül.




# Geometriai algoritmusok

A konvex burok előállítása nélkül, gyorsan (lineáris időben) kereshető három olyan A,B,C pont, hogy $Q \in (A,B,C)$.

Legyen $A{:=}p_1$, majd keressünk olyan B pontot, hogy A, B és Q nem esik egy egyenesre! Ha nincs ilyen B pont, akkor nincs megoldás.

Ezután minden további P pontra határozzuk meg, hogy a (Q,A) és (Q,B) egyenesek által meghatározott síknegyedek melyikébe esik P.

Nincs megoldás, ha minden pont (Q,A)-tól balra és (Q,B)-től jobbra van.

# Geometriai algoritmusok

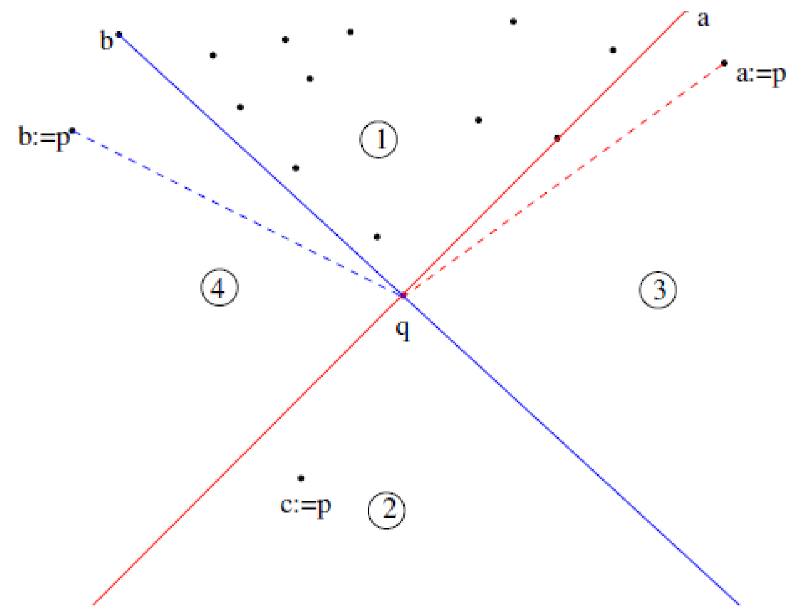
Az eddig vizsgált pontok a (B,Q,A) tartományba esnek. Újabb p pont esetén:

1. eset. Fordul(Q,A,p)≤0 és Fordul (Q,B,p)≥0: nem módosul semmi.

2. eset. Ha Fordul (Q,A,p)≥0 és Fordul (Q,B,p)≤0: C:=p

3. eset. Ha Fordul (Q,A,p)>0 és Fordul (Q,B,p)>0: A:=p.

4. eset. Ha Fordul (Q,B, p)<0 és Fordul (Q,A, p)<0): B:=p.




# Geometriai algoritmusok

```
Kerítés(A,B,C,N,Q):
  A:=1; i:=2
  Ciklus amíg i≤N és Fordul(Q,P(A),P(i))=0
    i:=i+1
  Ciklus vége
  Ha i>N akkor B:=0, C:=0
        {P pontjai a Q-val egy egyenesen vannak}
  különben B:=i; C:=0
           Ha Fordul(Q,P(A),P(B))>0
                                  akkor A:=B; B:=1
           {(Q,P(A))-tól P(B) balra legyen}
           …
```

# Geometriai algoritmusok

```
Ciklus amíg j=1-től N-ig
   fqa:=Fordul(Q,P(A),P(j))
   fqb:=Fordul(Q,P(B),P(j))
   Ha fqa≥0 és fqb≤0 akkor C:=j; Kilépés
                                          {2. eset}
   Ha fqa>0 és fqb>0 akkor A:=j  {3. eset}
   különben ha fqa<0 és fqb<0 akkor B:=j
                                          {4. eset}
Ciklus vége
{ha C=0, akkor Q kívül van, és ekkor (Q,P(A))
 és (Q,P(B)) érintő egyenes}
 Ha C>0 akkor … {az (A,B,C) háromszög
                            tartalmazza Q-t}
```

# Geometriai algoritmusok

```
{az (A,B,C) háromszög tartalmazza Q-t}
Ciklus i=1-től N-ig
   Ha i≠A és i≠B és i≠C és
      Belül(P(A),P(B),P(C),P(i))
   akkor fqa:=Fordul(Q,P(A),P(i))
         fqb:=Fordul(Q,P(B),P(i))
         fqc:=Fordul(Q,P(C),P(i))
         Ha Fordul(Q,P(A),P(B))=0 és
            Fordul(P(i),P(A),P(B))=0) {AB oldal}
            akkor Ha fqc<0 akkor A:=i
                           különben B:=i
         különben ha …
```

Piros kód: Q az (A,B,C) háromszög valamelyik oldalán van.

# Geometriai algoritmusok

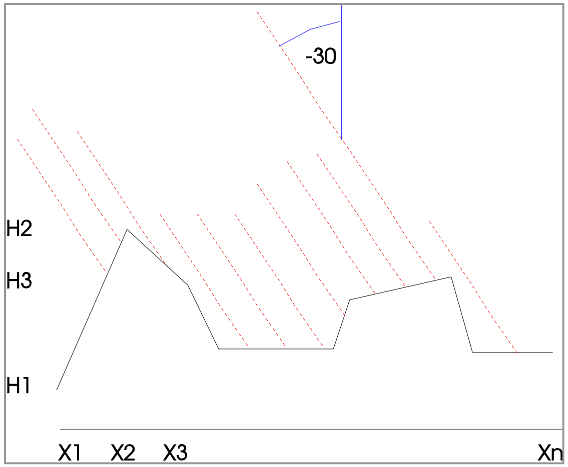
```
  különben ha Fordul(Q,P(B),P(C))=0 és
              Fordul(P(i),P(B),P(C))=0 {BC oldal}
                 akkor ha fqa<0 akkor B:=i
                             különben C:=i
  különben ha Fordul(Q,P(C),P(A))=0 és
              Fordul(P(i),P(C),P(A))=0 {CA oldal}
                 akkor ha fqb<0 akkor C:=i
                             különben A:=i
  különben ha fqb≥0 és fqc≤0 akkor A:=i
  különben ha fqa≤0 és fqc≥0 akkor B:=i
                          különben C:=i
  Ciklus vége
Eljárás vége.
```

# Feladat:

Egy látképet egyenes szakaszok sorozatával adunk meg. A látkép felett a függőleges iránnyal az óramutató járása szerint α szöget bezárva, végtelen távolságban van a Nap.

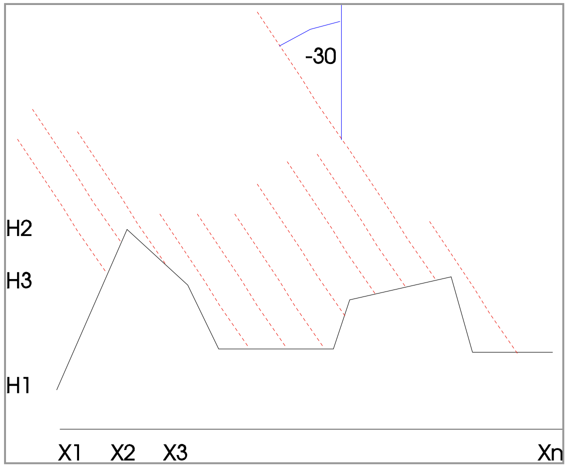
Add meg, hogy a Nap megvilágítja-e a teljes látképet!

Ha nem, akkor add meg a megvilágítás irányából az első olyan szakasz sorszámát, amelyet a Nap nem világít meg!



# Geometriai algoritmusok

```
Árnyék(N,H,Mind,E,Alfa):
   (DX,DY):=Alfa szögből számítva
   i:=1
   Ciklus amíg i<N és
               Fordul(H(i)-(DX,DY),H(i),H(i+1))<0
      i:=i+1
   Ciklus vége
   Mind:=i≥N
   Ha nem Mind akkor E:=i
Eljárás vége
```




# Geometriai algoritmusok

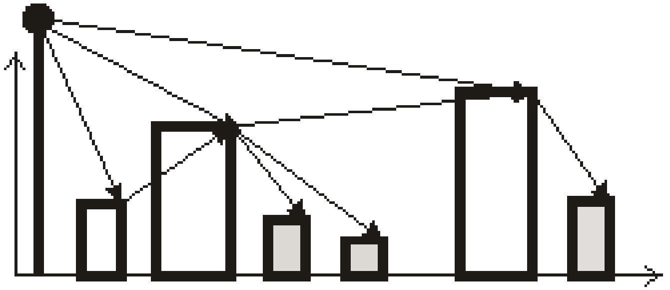
**Feladat:**

Egy koordinátarendszerben az x-tengely mentén téglalap alakú házak helyezkednek el. Egy lámpa balról, fentről világítja meg a házakat.

Melyek azok a házak, amelyek teljesen árnyékban vannak?

**Megjegyzés:**

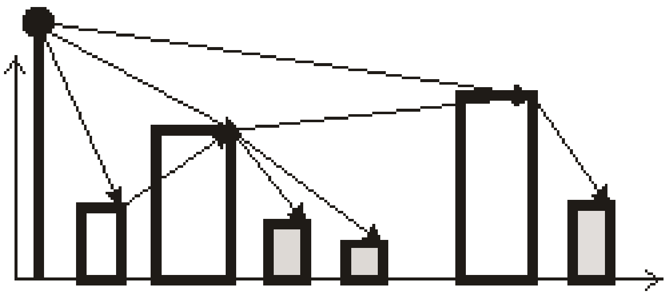
Az ábra szerint elég a házak bal felső sarkát ismerni!



## Megoldás:

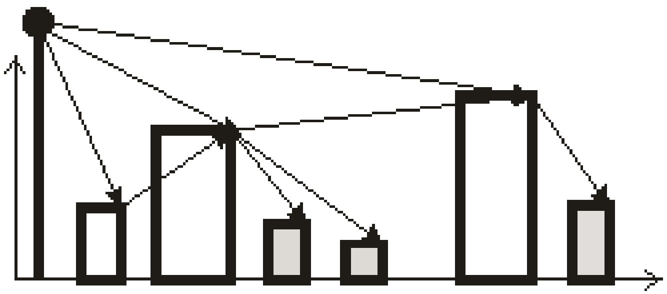
Legyen L a lámpa, H(i) az i-edik ház jobb felső sarkának helye, u pedig az utoljára megvilágított ház!

Azok a házak vannak árnyékban, amelyeket az előttük levő utolsó megvilágított ház teljesen takar. Azaz a H→H(u)→H(i) úton nem balra kell fordulni!



# Geometriai algoritmusok

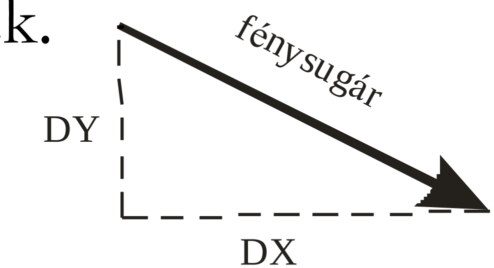
```
Árnyék(L,N,H,Db,Y):
  Db:=0; u:=1
  Ciklus i=2-től N-ig
    Ha Fordul(L,H(u),H(i))<0
          akkor u:=i
      különben Db:=Db+1; Y(Db):=i
  Ciklus vége
Eljárás vége
```



# Geometriai algoritmusok

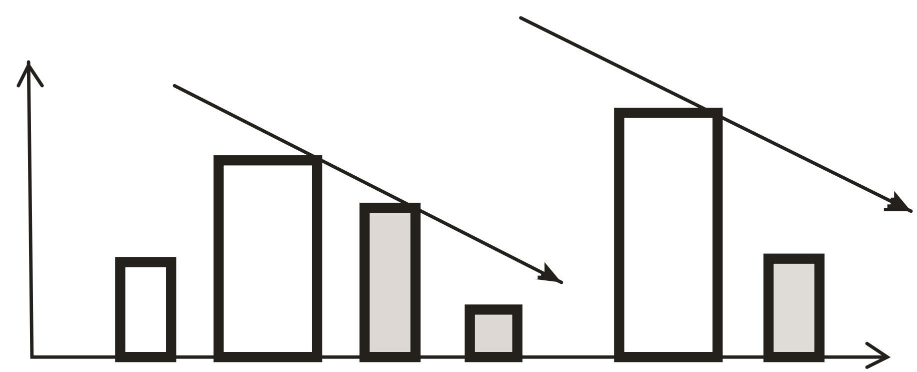
## Feladat:

Egy koordinátarendszerben az x-tengely mentén téglalap alakú házak helyezkednek el. A Nap balról, fentről süt rájuk, a házakhoz párhuzamos fénysugarak érkeznek.




Melyek azok a házak, amelyeknek legalább egyetlen pontjára süt a nap?

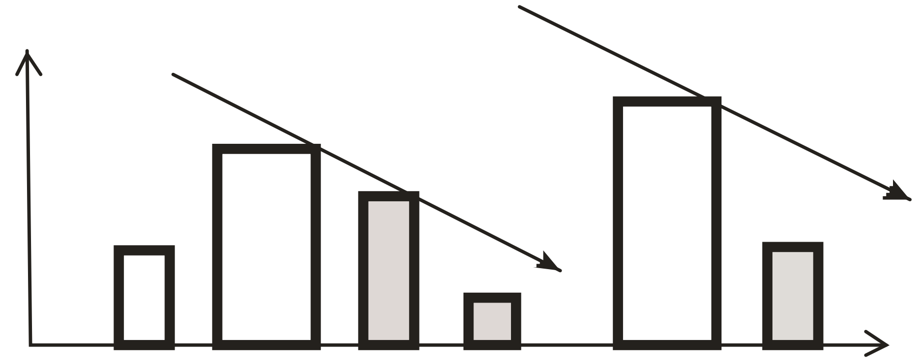
Az ábra szerint elég a házak bal felső sarkát ismerni!



## Megoldás:

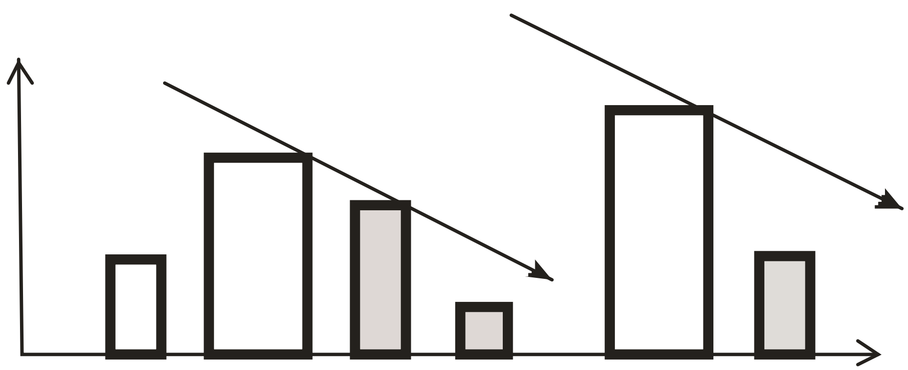
Legyen H(i) az i-edik ház jobb felső sarkának helye, u pedig az utoljára megvilágított ház!

Azok a házak vannak megvilágítva, amelyeket az előttük levő utolsó megvilágított ház nem takar. Azaz a H→H(u)→H(i) úton balra kell fordulni!



# Geometriai algoritmusok

```
Világos(DX,DY,N,H,Db,Y):
  Db:=1; Y(Db):=1
  Ciklus i=2-től N-ig
    Ha irány(H(Y(Db))-(DX,DY),H(Y(Db)),H(i))<0
          akkor Db:=Db+1; Y(Db):=i
  Ciklus vége
Eljárás vége
```



# Geometriai algoritmusok

## Feladat:

Adott a síkon n darab pont, amelyek nem esnek egy egyenesre. A pontok (x,y) koordinátáikkal adottak. Kössünk össze pontpárokat egyenes szakaszokkal úgy, hogy olyan zárt poligont kapjunk, amelyben nincs metsző szakaszpár!

# Geometriai algoritmusok

## Megoldás:

Válasszuk ki a legkisebb x-koordinátájú pontot, ha több ilyen van, akkor ezek közül válasszuk a legkisebb y-koordinátájút! Ezt nevezzük (bal-alsó) sarokpontnak és cseréljük meg az első ponttal!

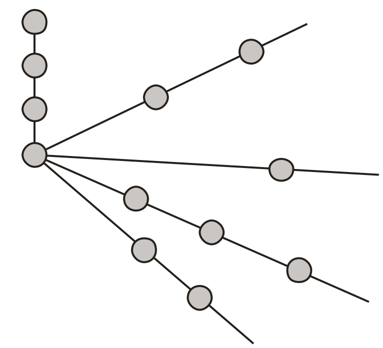
```
Poligon(N,P):
  Sarokpont(N,P); Rendez(N,P); Fordít(N,P)
Eljárás vége.
```

# Geometriai algoritmusok

```
Sarokpont(N,P):
  s:=1
  Ciklus i=2-től N-ig
    Ha P(i).x<P(s).x vagy
       P(i).x=P(s).x és P(i).y<P(s).y
     akkor s:=i
  Ciklus vége
  Csere(P(1),P(s))
Eljárás vége.
```

# Geometriai algoritmusok

Húzzunk (fél) egyenest a sarokpontból minden ponthoz! Rendezzük az egyeneseket a sarokponton áthaladó, x-tengellyel párhuzamos egyenessel bezárt (előjeles) szög alapján!



# Geometriai algoritmusok

A sarokpont legyen az első, és $p_i$ előbb legyen mint $p_j$ akkor és csak akkor, ha a $p_1 \rightarrow p_i \rightarrow p_j$ úton balra kell fordulni, vagy nem kell fordulni, de $p_j$ van közelebb a $p_1$ -hez!

```
kisebb(Q,a,b):
  ir:=Fordul(Q,a,b)
  kisebb:=ir=-1 vagy ir=0 és a.x<b.x
Eljárás vége.
```

A rendezésnél persze elveszik a pontok régi sorszáma, azaz célszerű azt megőrizni, pl. a PONT típusba egy újmezőt, a sorszám mezőt felvéve.

# Geometriai algoritmusok

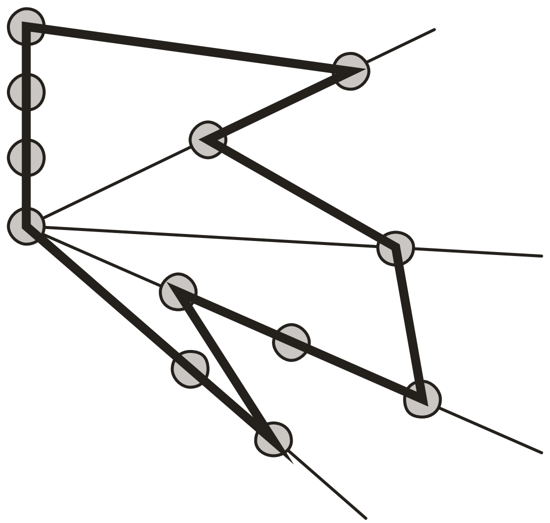
```
Rendez(e,u,P):
  i:=e; j:=u; s:=P(i);
  Ciklus amíg i<j
    Ciklus amíg i<j és nagyobb(P(1),P(j),s)
      j:=j-1
    Ciklus vége
    Ha i<j akkor
      P(i):=P(j); i:=i+1
      Ciklus amíg i<j és nagyobb(P(1),s,P(i))
        i:=i+1
      Ciklus vége
      Ha i<j akkor P(j):=P(i); j:=j-1
  Ciklus vége
  P(i):=s
  Ha i-e>1 akkor Rendez(e,i-1,P);
  Ha u-i>1 then Rendez(i+1,u,P);
Eljárás vége.
```

# Geometriai algoritmusok

```
Fordít(N,P):
  Ha N>2 és Fordul(P(1),P(N),P(N-1))=0
    akkor Verembe(P(N)); Fordít(N-1,P)
          Veremből(P(N))
Eljárás vége.
```

Kérdés: Mit csinált a rendezés, ha az utolsó irány a függőleges, azaz a pontok x-koordinátája megegyezik?

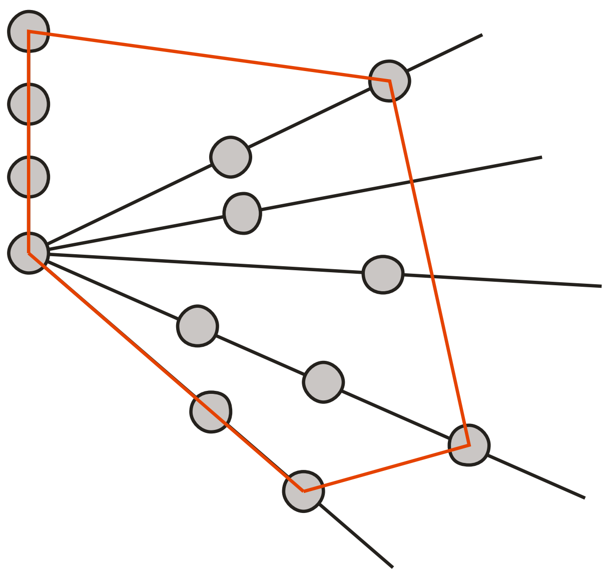
Kössük össze a pontokat a kapott sorrendben!



# Geometriai algoritmusok

## Feladat:

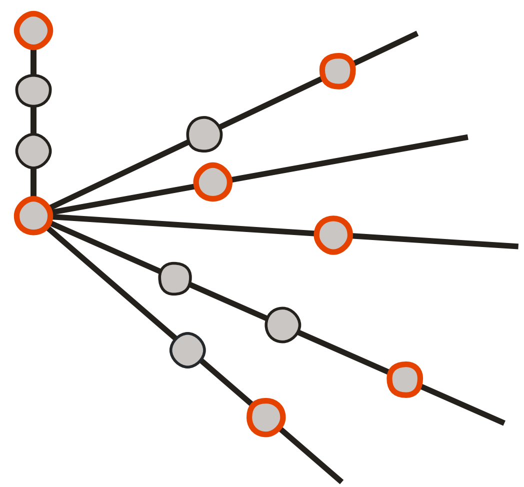
Adott a síkon n darab pont, amelyek nem esnek egy egyenesre. A pontok (x,y) koordinátáikkal adottak. Adjuk meg a legkisebb konvex poligont, amely az összes pontot tartalmazza!



# Geometriai algoritmusok

## Megoldás:

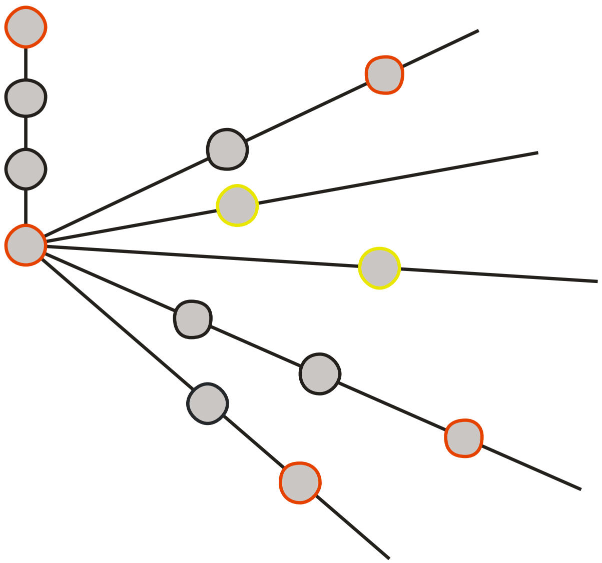
Első lépésként rendezzük a ponthalmazt a bal-alsó sarokpontra vett polárszög szerint, majd minden egyenesen csak a sarokponttól legtávolabbi pontot hagyjuk meg, a többit töröljük. Az így törölt pontok biztosan nem lehetnek csúcspontjai a konvex buroknak.



# Megoldás:

Második lépésként haladjunk körbe a megmaradt pontokon! Hagyjuk el a $q_{i+1}$ pontot, ha a $q_i \rightarrow q_{i+1} \rightarrow q_{i+2}$ úton nem balra kell fordulni!

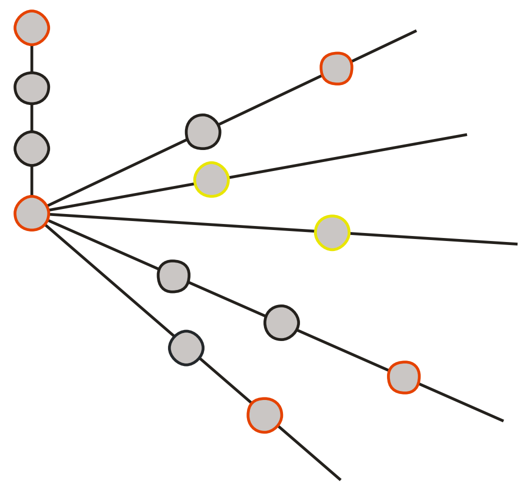
Megjegyzés: ez az elv a korábban ki-hagyandónak ítélt pontokra is műkö-dik, azaz nem kell kihagyással külön foglalkozni!



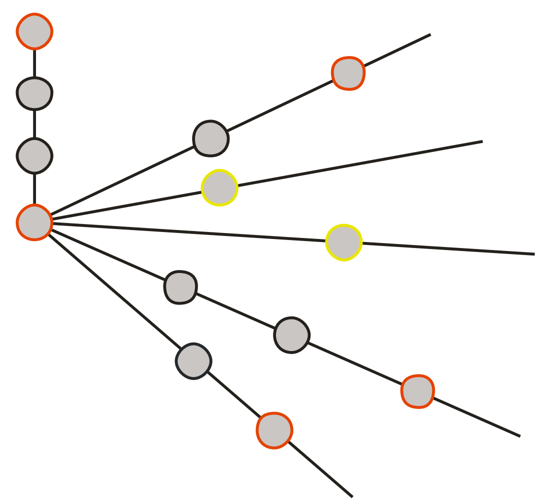
# Geometriai algoritmusok

A rendezésnél persze elveszik a pontok régi sorszáma, azaz célszerű azt megőrizni, pl. a PONT típusba egy újmezőt, a sorszám mezőt felvéve.

```
Konvex burok(N,P):
  Sarokpont(N,P); Rendez(N,P); Fordít(N,P)
  P(N+1):=P(1); Körbejár(N,P)
Eljárás vége.
```



```
Körbejár(N,P,B):
  i:=3
  Ciklus amíg Fordul(P(1),P(i-1),P(i))=0
    i:=i+1
  Ciklus vége
  B(1):=P(1); B(2):=P(i-1); M:=2
  Ciklus amíg i≤N+1
    Ha Fordul(P(B(M-1)),P(B(M)),P(i))≥0
          akkor M:=M-1
       különben M:=M+1; B(M):=I
    i:=i+1
  Ciklus vége
  M:=M-1
Eljárás vége.
```



## Fekete-fehér párosítás a síkon.

Adott a síkon n darab fehér és n darab fekete pont úgy, hogy bármely három pont nem esik egy egyenesre. Párosítani kell a fehér pontokat fekete pontokkal úgy, hogy a párokat összekötő szakaszok ne metsszék egymást!

**Állítás**: Létezik olyan $p_i$ és $p_j$ különböző színű pontpár, hogy a $(p_i;p_j)$ egyenes mindkét oldalán a fehér pontok száma megegyezik a fekete pontok számával.

# Geometriai algoritmusok

Rendezzük a pontokat a bal-alsó sarokponthoz viszonyított polárszög szerint! Tegyük fel, hogy a rendezésben az első pont fekete! Jelölje $d_i$ az első i pont közül a feketék számából kivonva a fehérek számát! Tehát $d_1=1$, $d_{2n}=0$ és $d_{i+1}=d_i+1$, ha az i+1-edik pont fekete, egyébként $d_{i+1}= d_i -1$. Ha a rendezésben utolsó, azaz 2n-edik pont fehér, akkor az 1. és 2n. pontpár megoldás. Ha a 2n-edik pont fekete, akkor $d_{2n-1}=-1$, de mivel $d_1=1$ és $d_{i+1}=d_i\pm1$, így van olyan $1<i<2n-1$ index, hogy $d_i = 0$. Ha az i-edik pont nem fehér, akkor a keresést az [1;i-1] intervallumban kell folytatni, ami véges sok lépés után véget ér.

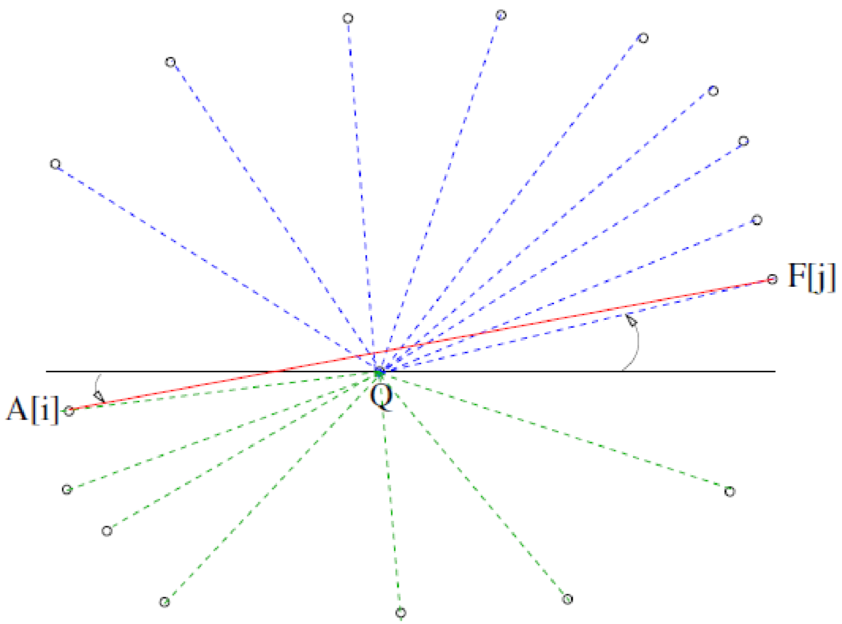
# Geometriai algoritmusok

```
Párosít(bal,jobb):
   Ha jobb=bal+1 akkor Kiír(bal,jobb)
   különben
      SarokPontRendez(P,bal,jobb); d:=1; i:=bal
      Ciklus amíg d≠0 vagy
                    nem(P(bal).az>n Xor P(i).az>n)
         i:=i+1
         Ha P(bal).az>n Xor P(i).az>n
              akkor d:=d-1 különben d:=d+1
      Ciklus vége
      Kiír(bal,i-1);
      Ha bal+1<i-1 akkor Párosít(bal+1, i-1)
      Ha i+1<jobb akkor Párosít(i+1,jobb)
Eljárás vége.
```

## Pontlefedés szakasszal

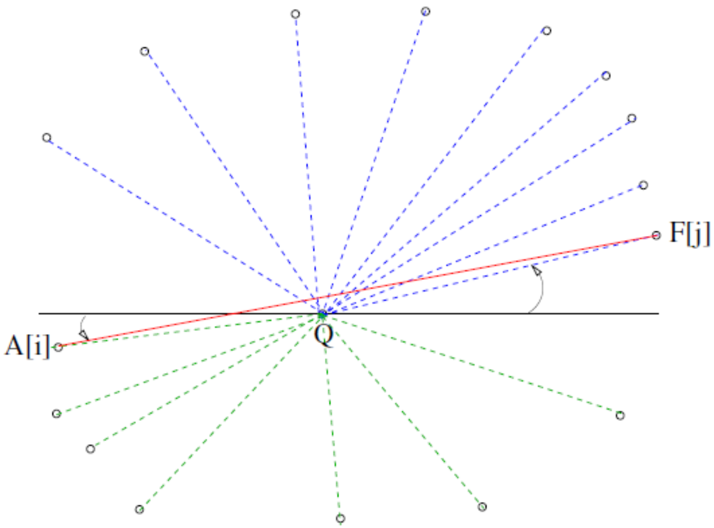
Adott a síkon egy P ponthalmaz és egy kitüntetett Q pontja. Kiszámítandó a P ponthalmaz két olyan A és B pontja, hogy a Q pont az A;B szakaszra esik.

**Követelmény**: O(n log n) futási idejű algoritmus.



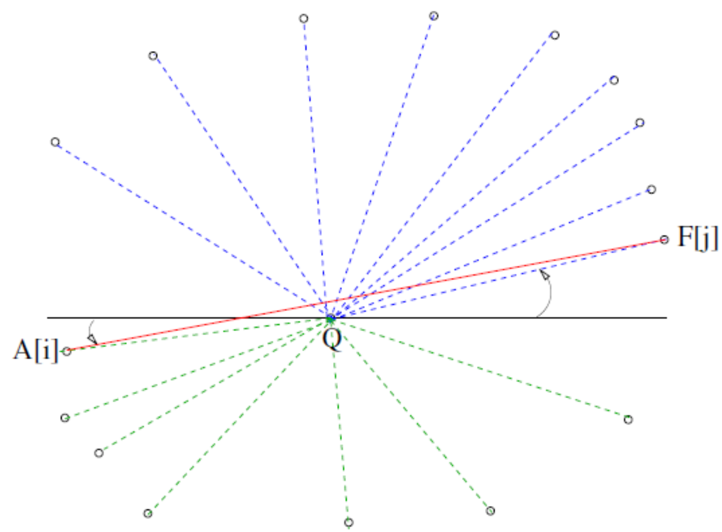

# Geometriai algoritmusok

Osszuk két diszjunkt részhalmazba P pontjait (már a beolvasáskor) aszerint, hogy a Q-n átmenő, x-tengellyel párhuzamos egyenes melyik oldalára esnek! Ha erre az egyenesre esik egy pont, akkor annak megfelelően, hogy Q előtt van-e. F-be kerüljenek az egyenes felettiek, A-be az alattiak! Majd rendezzük a két ponthalmazt a Q pontra vett polárszög szerint!



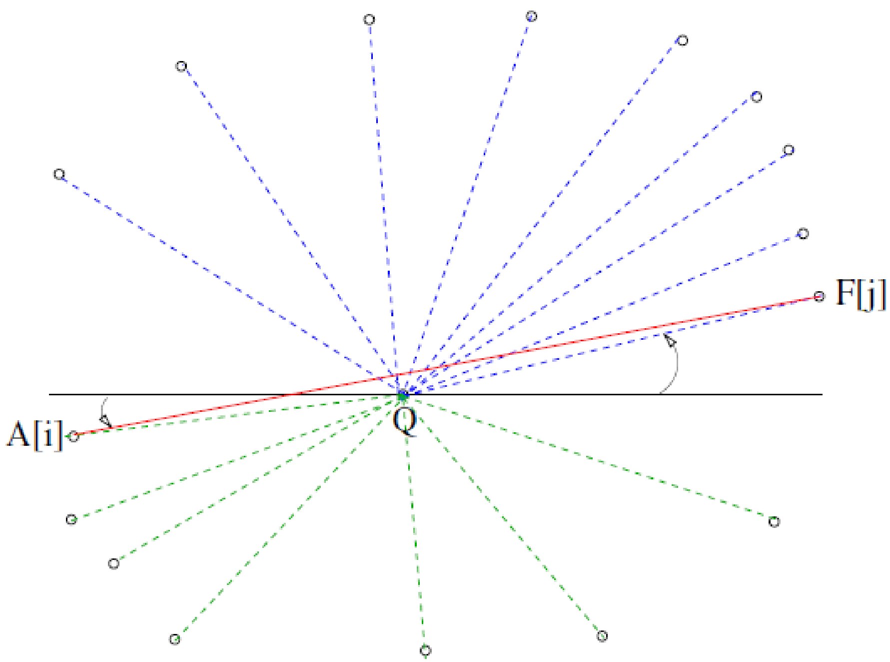

# Geometriai algoritmusok

Legyen i := 1; j := 1! Ha Q az (A(i),F(j))-től jobbra van, akkor az A(i)-hez nincs olyan pont F-ben, amely megoldás. Hasonlóan, ha Q az (A(i),F(j))-től balra van, akkor az F( j)-hez nincs olyan pont A-ben, amely megoldás. Tehát vagy i, vagy j növelhető és nem vesztünk el lehetséges megoldást, azaz minden ii<i és jj<j esetén a (A(ii),F(jj)) nem lehet megoldás!

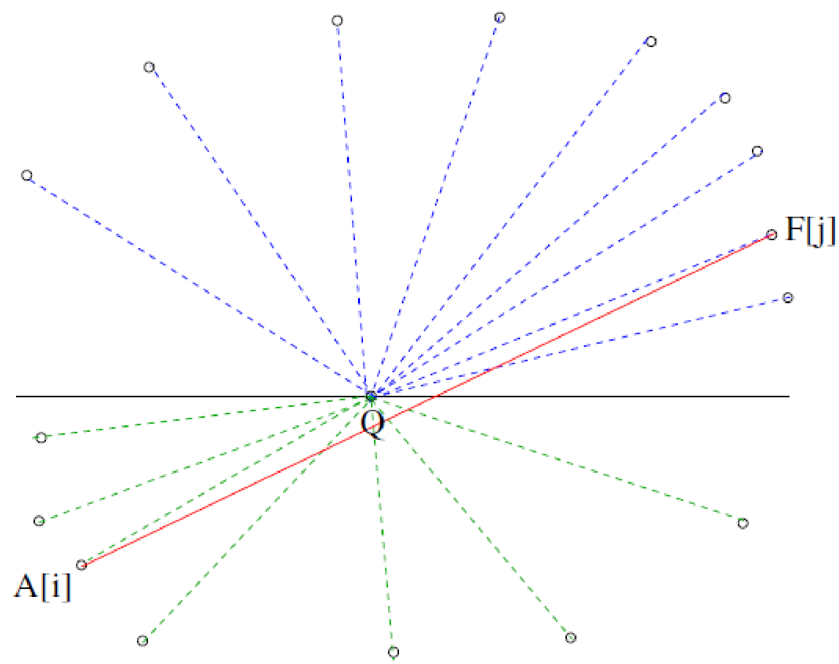



# Geometriai algoritmusok

i:=i+1




j:=j+1




# Geometriai algoritmusok

```
Pontlefedés(A,n1,B,n2,van,i,j):
  SarokRendez(A,n1); SarokRendez(F,n2)
  i:=1; j:=1; van:=hamis
  Ciklus amíg i≤n1 és j≤n2 és nem van
    FAFQ:=Fordul(A(i),F(j),Q)
    Ha FAFQ<0 akkor j:=j+1
    különben ha FAFQ>0 akkor i:=i+1
    különben van:=igaz
  Ciklus vége
Eljárás vége.
```

# Geometriai algoritmusok
# előadás vége