Geometriai algoritmusok

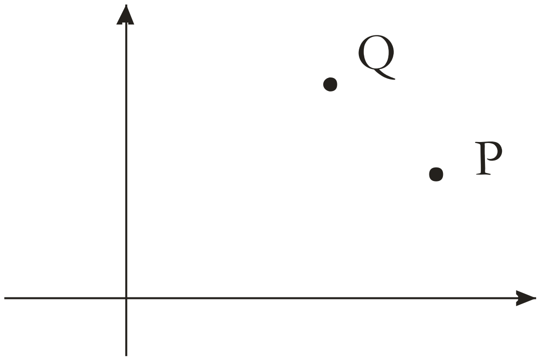
(Horváth Gyula és Szlávi Péter előadásai felhasználásával)

Geometriai algoritmusok

Feladat:

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

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



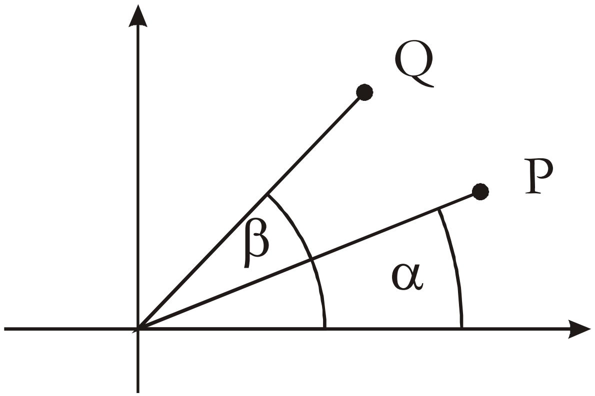
Ponttípus:

Típus Tpont=rekord(x,y: Egész)

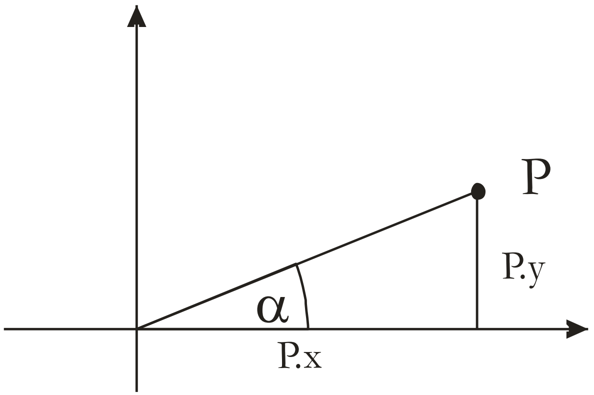
Geometriai algoritmusok

Értelmezés:

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



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



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

$\tan(\beta) = Q.y / Q.x$

Geometriai algoritmusok

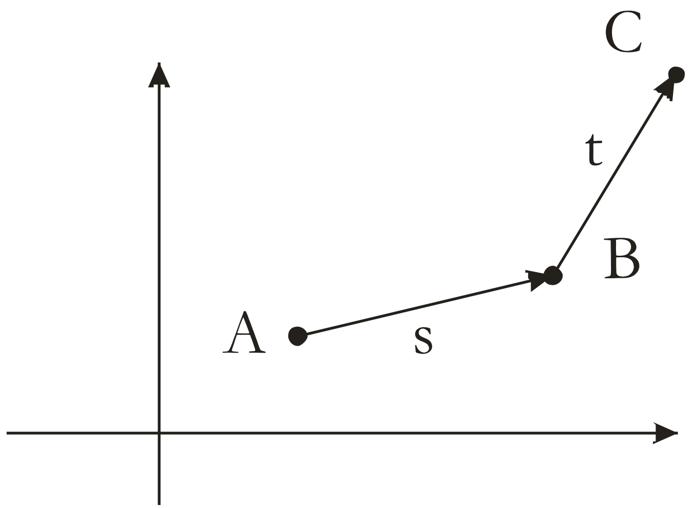
$\alpha<\beta \leftrightarrow \tan(\alpha)<\tan(\beta) \leftrightarrow P.y/P.x<Q.y/Q.x \leftrightarrow$
$P.y*Q.x<Q.y*P.x \leftrightarrow P.y*Q.x-Q.y*P.x<0$

```
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:

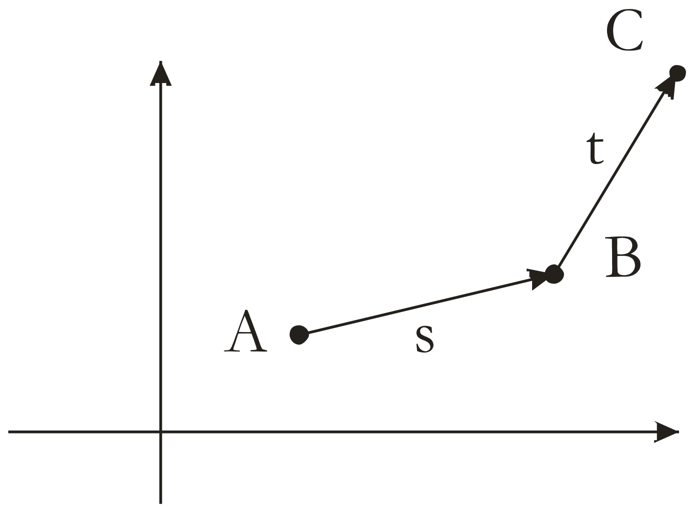
Toljuk el az **s**-t és a **t**-t úgy, hogy az A pont az origóba kerüljön! Ezzel visszavezetjük az „irányos” feladatra!

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

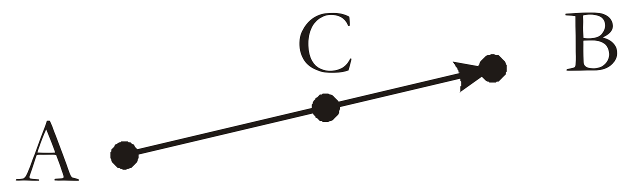
```
Fordul(A,B,C):
  P:=B-A          {P.x:=B.x-A.x; P.y:=B.y-A.y}
  Q:=C-A          {Q.x:=C.x-A.x; Q.y:=C.y-A.y}
  Fordul:=Irány(P,Q)
Függvény vége.
```



Geometriai algoritmusok

Feladat:

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

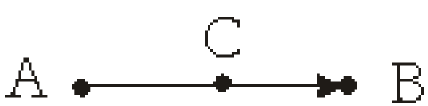


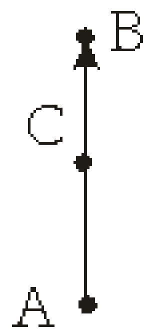
Megoldás:

- 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.
```



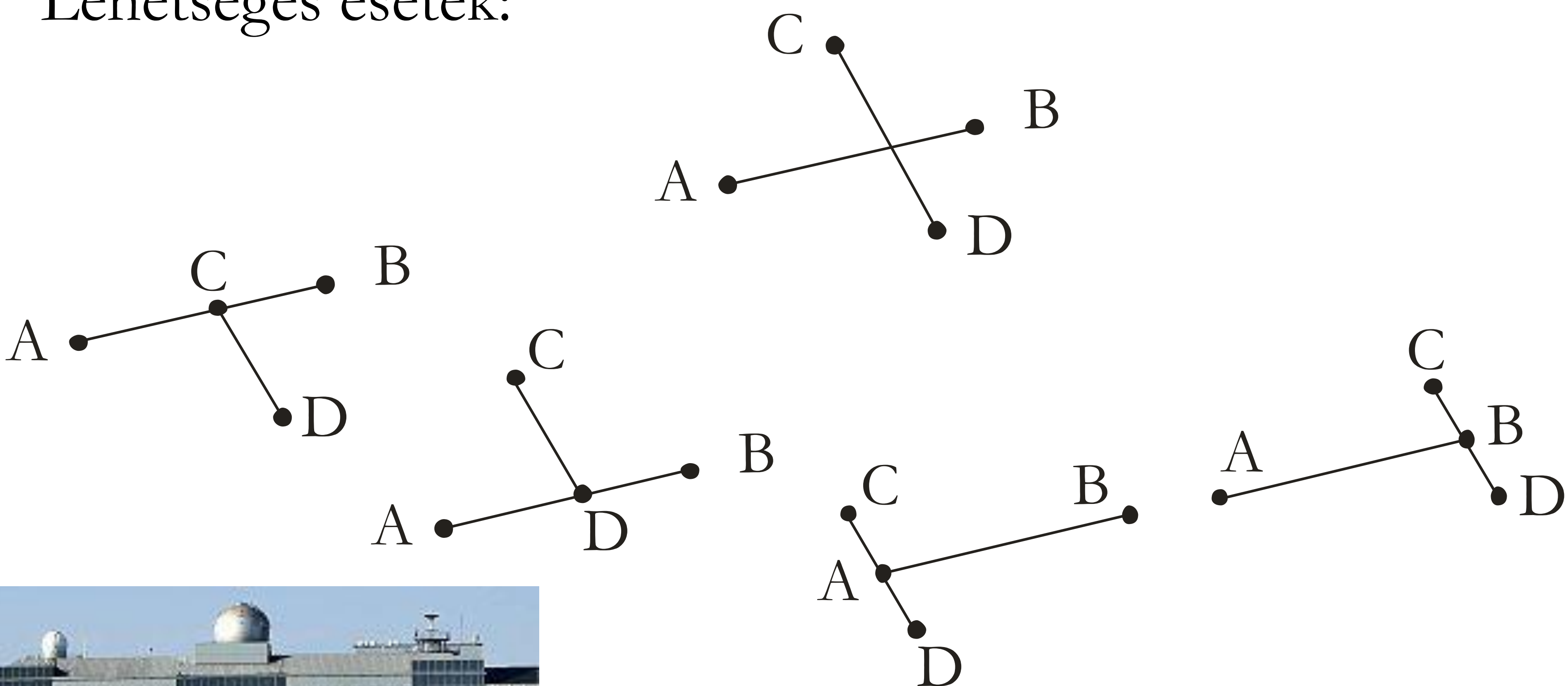


Geometriai algoritmusok

Feladat:

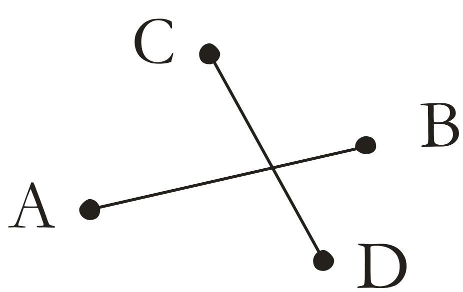
Döntsük el, hogy az (A,B) szakasz metszi-e a (C,D) szakaszt!

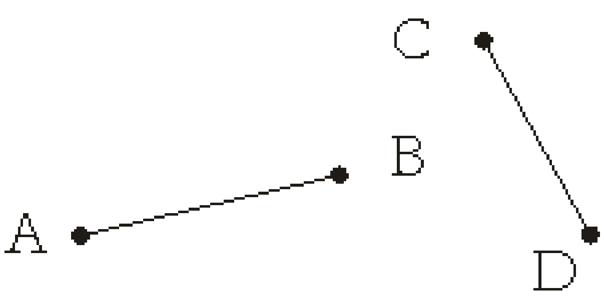
Lehetséges esetek:



Geometriai algoritmusok

```
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.
```





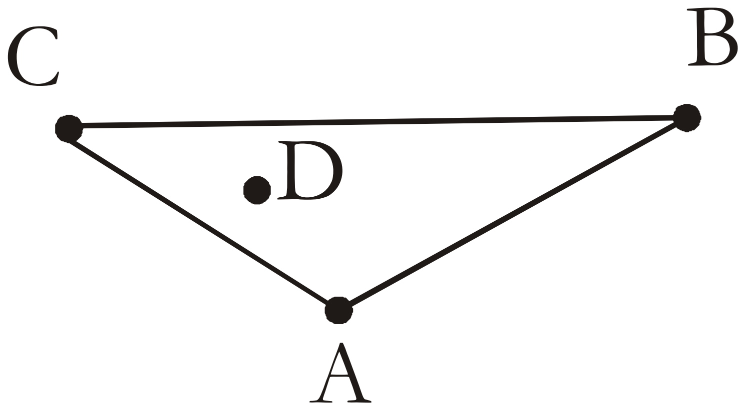
Geometriai algoritmusok

Feladat:

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

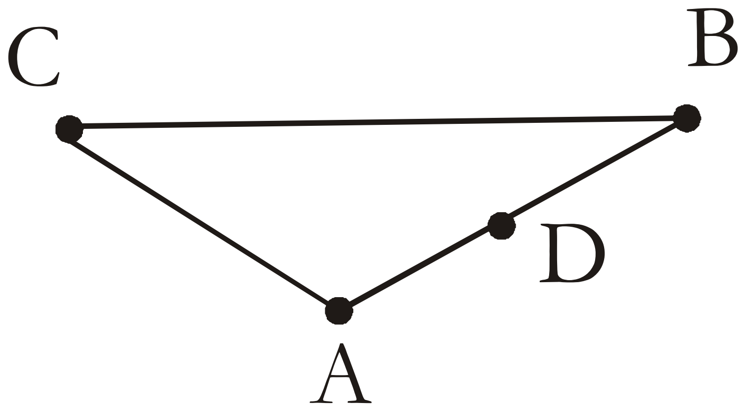
Megoldásötlet:

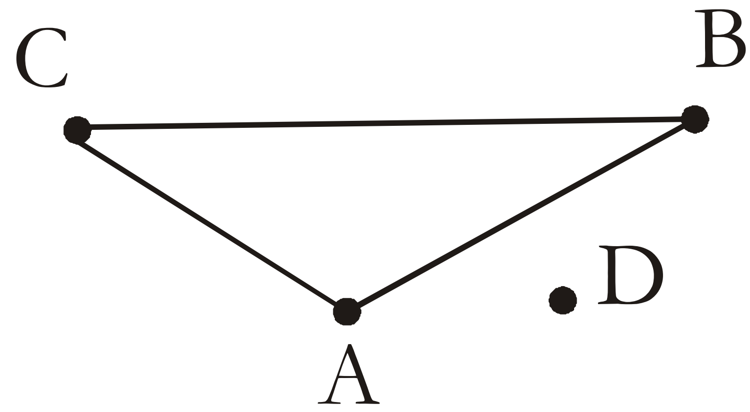
Belül van, ha a háromszöget A→B→C→A sorrendben körbejárva a D pont vagy mindig balra, vagy mindig jobbra van.



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.
```





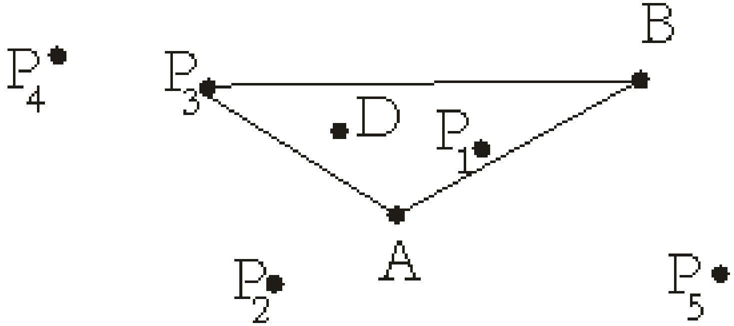
Geometriai algoritmusok

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) háromszög belsejében legyen!

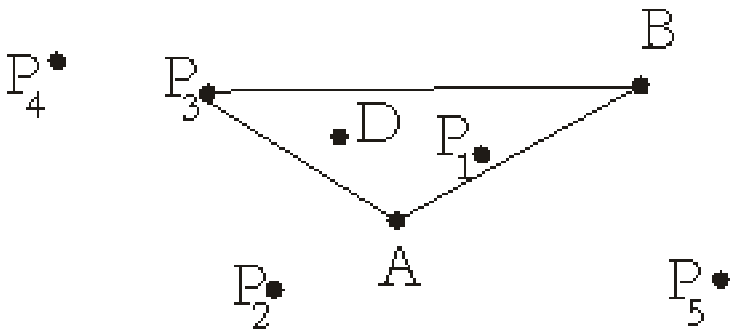
Megoldásötlet:

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

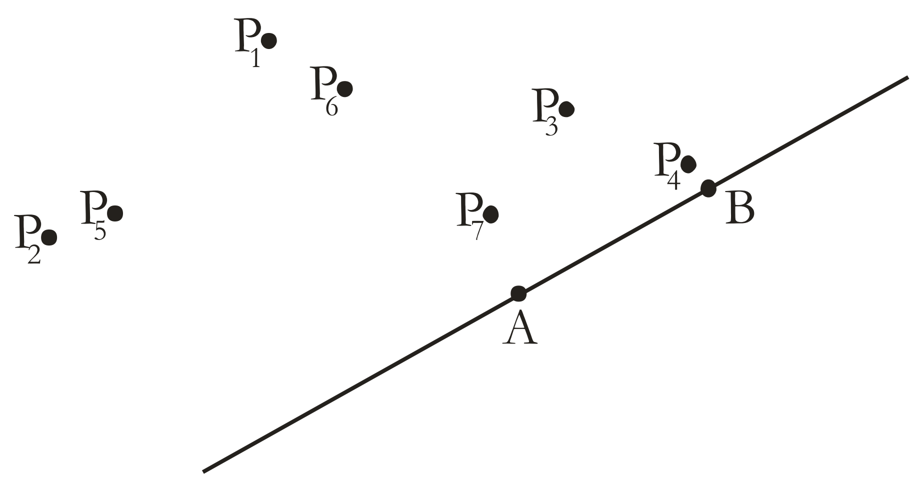


Geometriai algoritmusok

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!

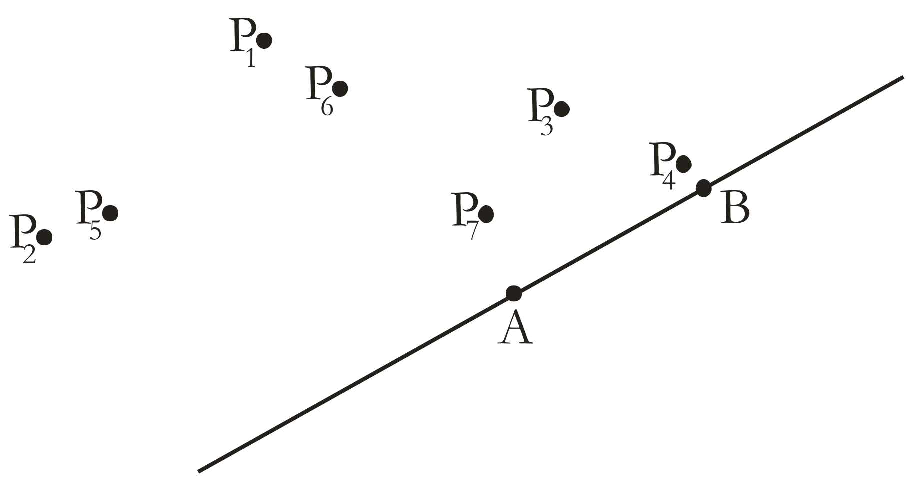
Feltehető, hogy az összes pont az (A,B) egyenestől balra van!



Geometriai algoritmusok

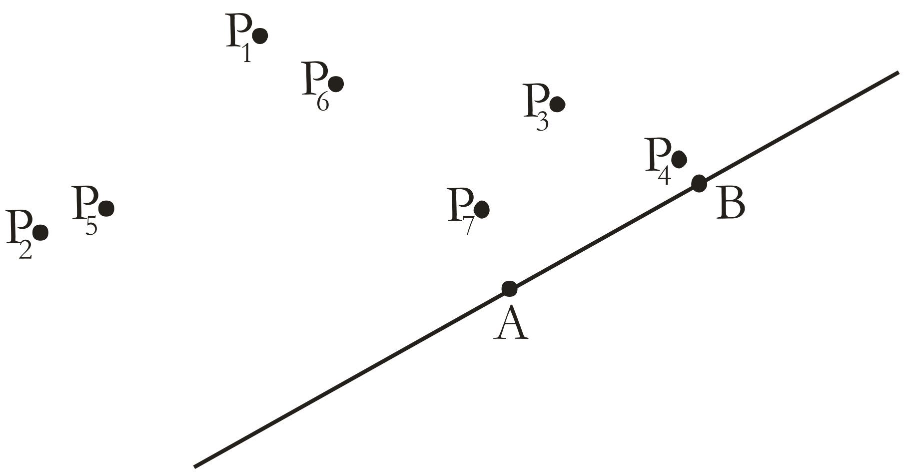
Megoldás:

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_5) háromszögben! Az (A,P_1)-től jobbra levők és a (B,P_1)-től balra levők pedig benne vannak.



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            {jobbra}
       Fordul(B,P(S),P(i))=-1              {balra}
     akkor S:=i
  Ciklus vége
Eljárás vége.
```

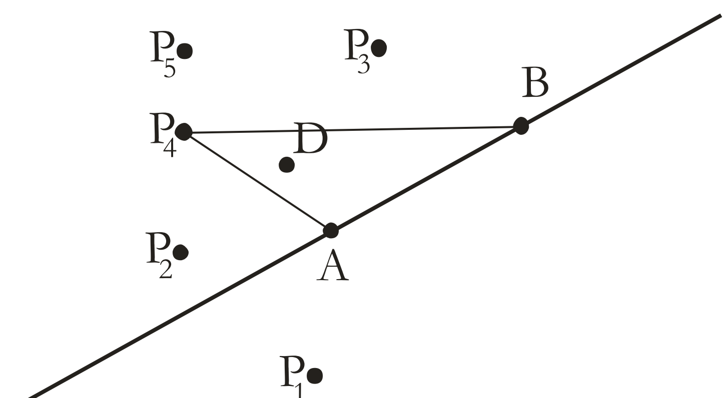


Geometriai algoritmusok

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!



Geometriai algoritmusok

Megoldás:

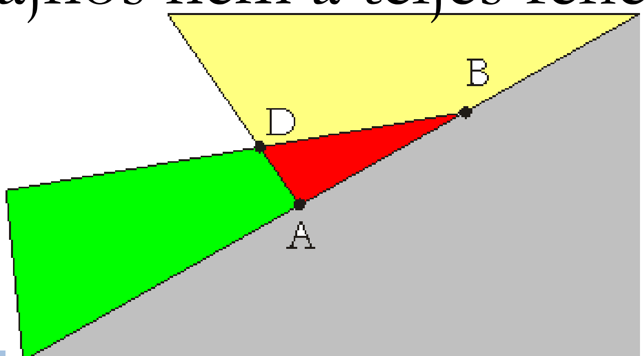
Nem oldható meg, ha a piros színű tartományban van pont.

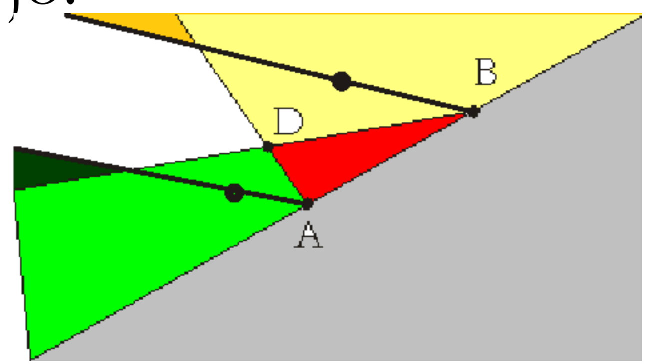
Az (A,B)-től jobbra levő pontok nem jók – szürke tartomány.

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!





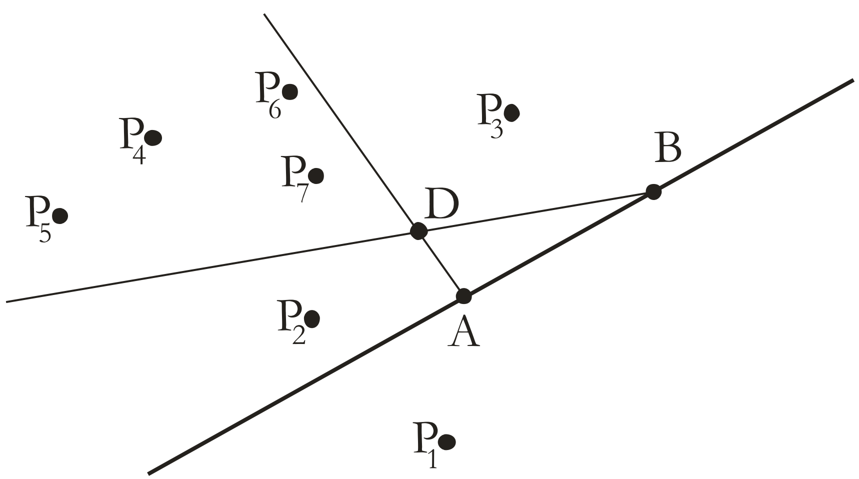
Geometriai algoritmusok

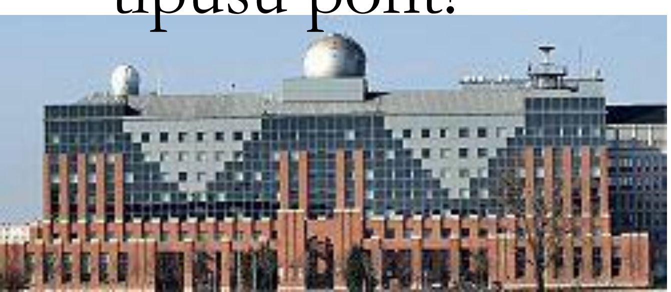
Megoldás:

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!

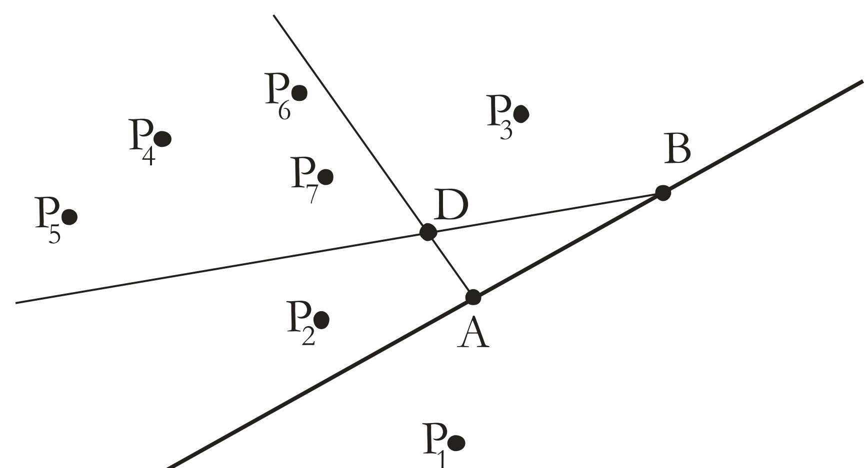
Finomíthatjuk a megoldást, ha feltesszük, hogy nincs P_2 és P_3 típusú pont!





Geometriai algoritmusok

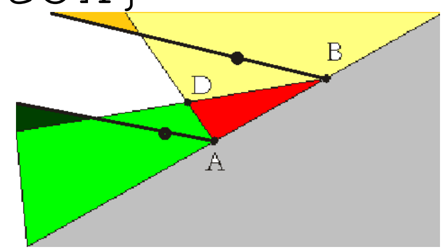
```
Keresés(A,B,N,P,D,Van,S):  {(A,B)-től és (A,D)-től balra}
  S:=0                     {(B,D)-től jobbra jelöltek}
  Ciklus i=1-től N-ig
    Ha Fordul(A,B,P(i))=-1 és Fordul(B,D,P(i))=1
             és Fordul(A,D,P(i))=-1
       akkor Ha S=0 akkor S:=i
             különben Ha Fordul(A,P(S),P(i))=1 és
                         Fordul(B,P(S),P(i))=-1
                         akkor S:=i
  Ciklus vége
  Van:=S>0
Eljárás vége.
```



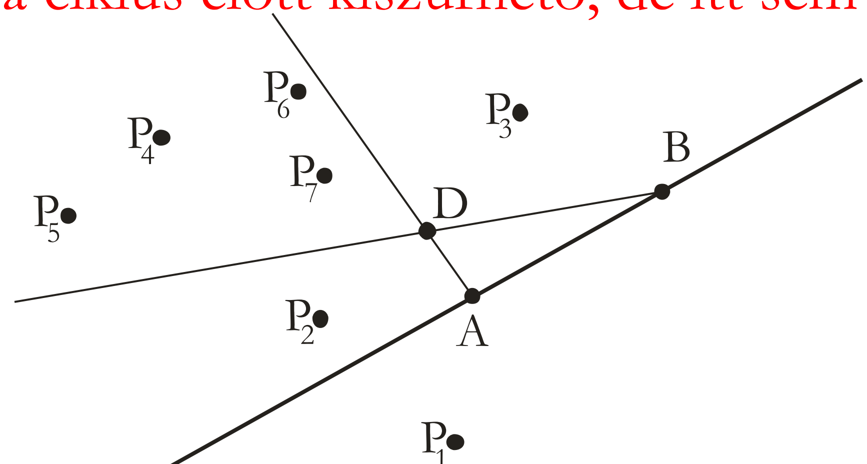
Geometriai algoritmusok

Ha van P_2 és P_3 típusú pont!

```
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 {zöld-sárga pontok}
    FAB:=Fordul(A,B,P(i))
    FBD:=Fordul(B,D,P(i))
    FDA:=Fordul(D,A,P(i))
```



{a szürke és a piros tartomány már a ciklus előtt kiszűrhető, de itt sem zavarnak}



Geometriai algoritmusok

Ha FAB<0 és FDA>0 és FBD<0 akkor {zöld}

  Ha AA=0 vagy Fordul(A,P(AA),P(i))>0

    akkor AA:=i {(A,B)-től balra, (D,A)-tól jobbra, (B,D)-től balra, (A,P(AA)-tól jobbra}

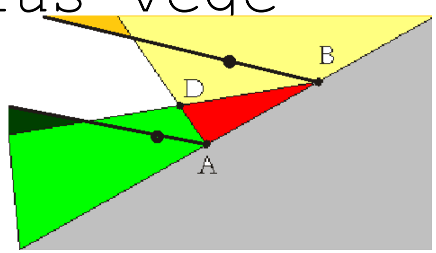
Ha FAB<0 és FDA<0 és FBD>0 akkor {sárga}

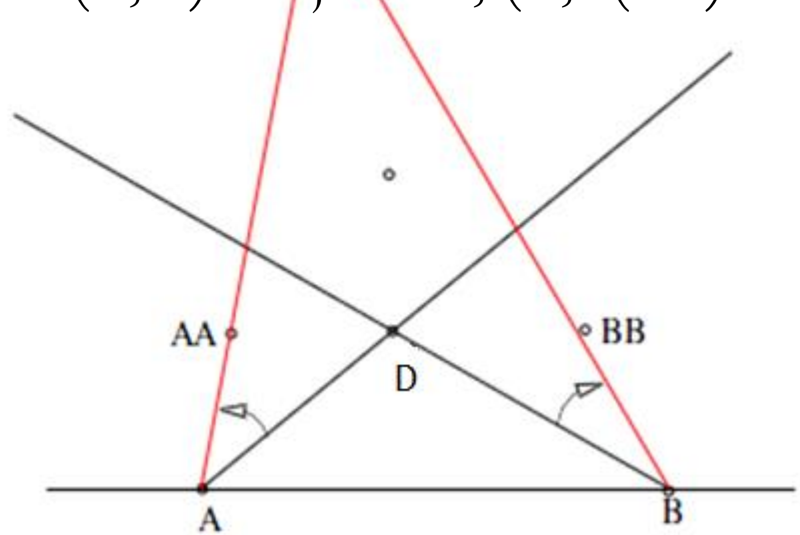
  Ha BB=0 vagy Fordul(B,P(BB),P(i))<0

    akkor BB:=i {(A,B)-től balra, (D,A)-tól balra, (B,D)-től jobbra, (B,P(BB)-tól balra}

Ciklus vége

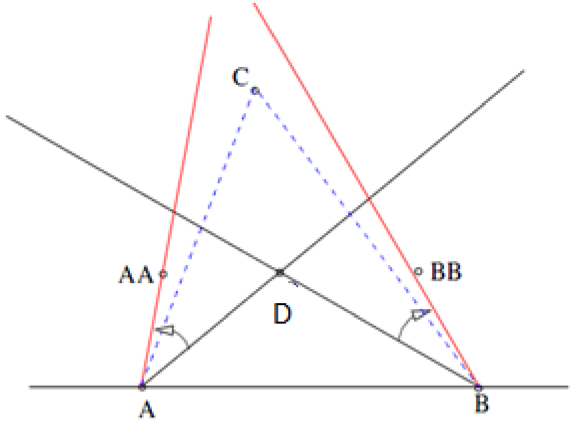
…





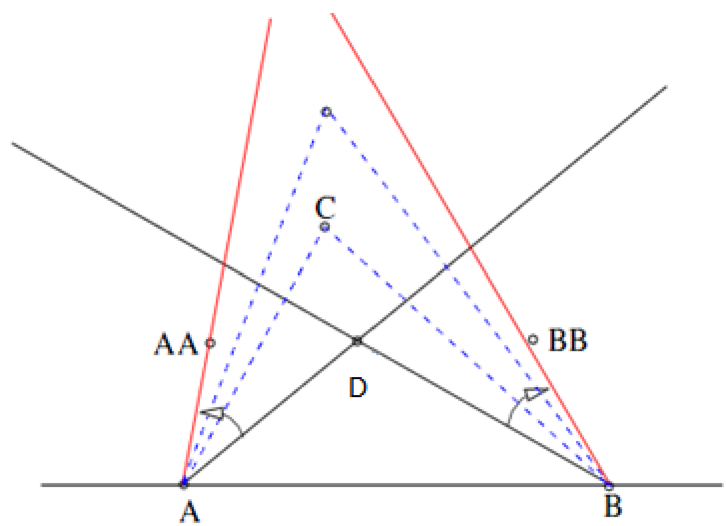
Geometriai algoritmusok

```
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     {(A,B)-től balra, (A,C)-től jobbra}
    Ha i≠AA és i≠BB akkor {(B,C)-től balra}
       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.
```



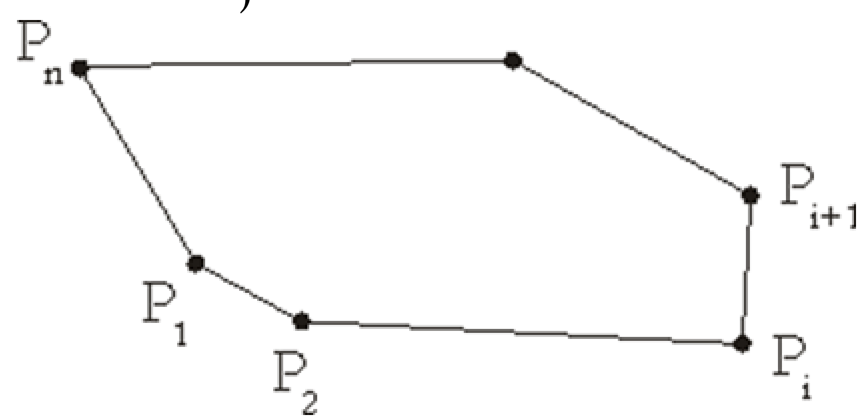
Geometriai algoritmusok

Feladat:

Döntsük el, hogy a $(P_1,\ldots,P_n)$ sokszög konvex sokszög-e! (A pontokat az óramutató járásával ellenkező sorrendben adjuk meg.)

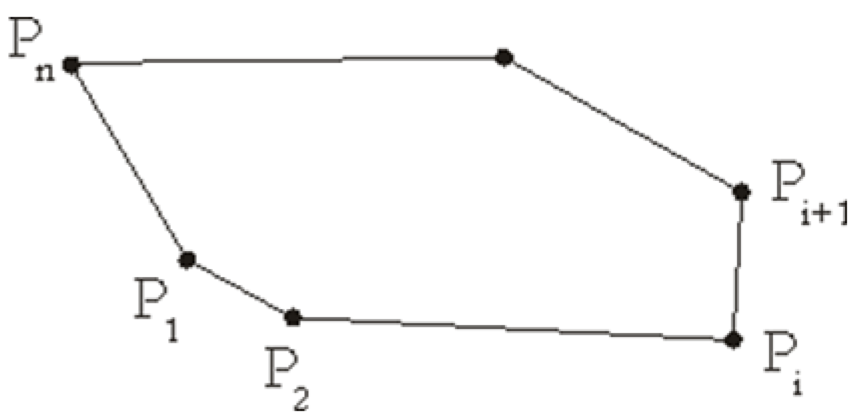
Megoldás:

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



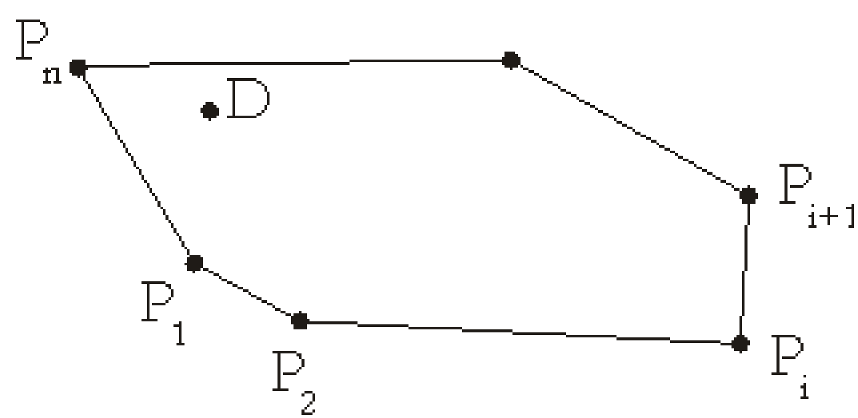
Geometriai algoritmusok

Feladat:

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

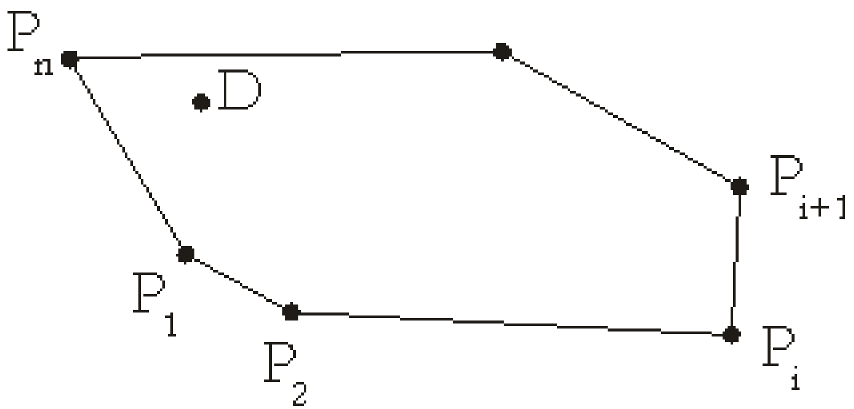
Megoldásötlet:

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.
```



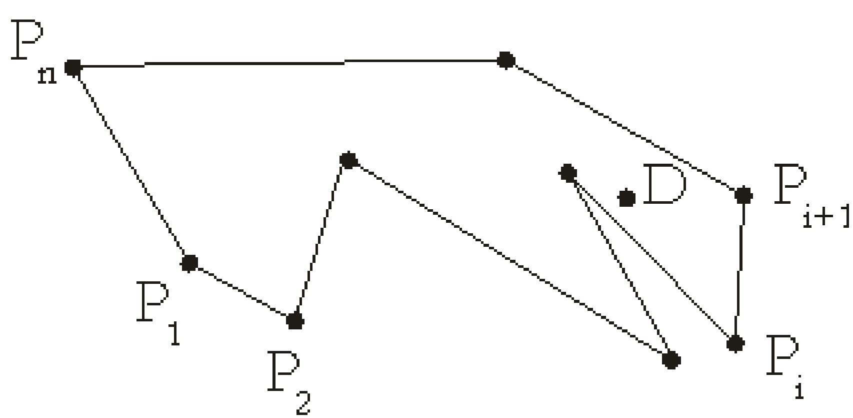
Geometriai algoritmusok

Feladat:

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

Probléma:

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



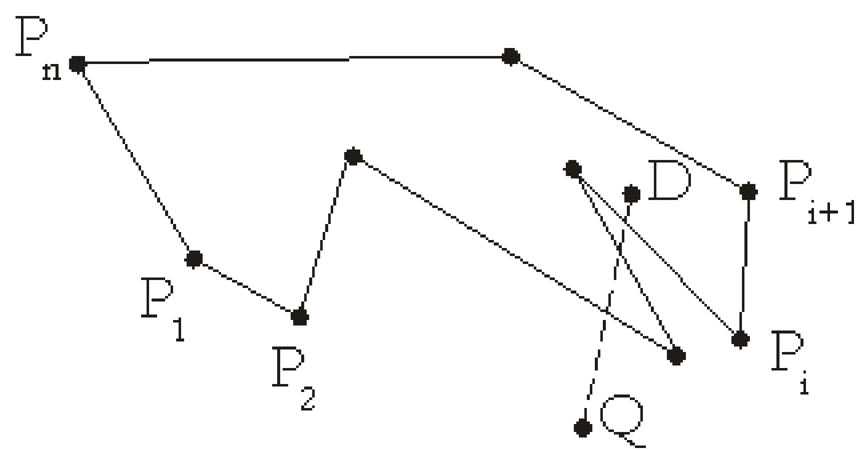
Geometriai algoritmusok

Megoldás:

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

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

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



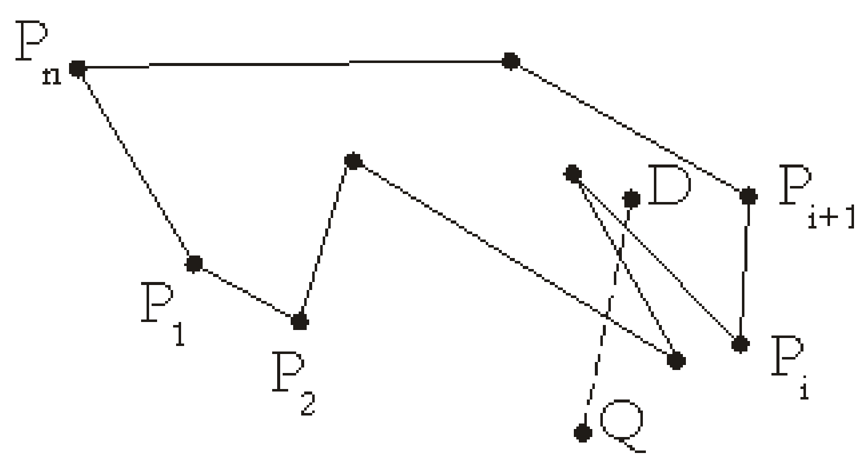
Geometriai algoritmusok

```
Külső(N,P,D,Q):
  Q.y:=P(1).y; Q.x:=-∞
  Ciklus i=1-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

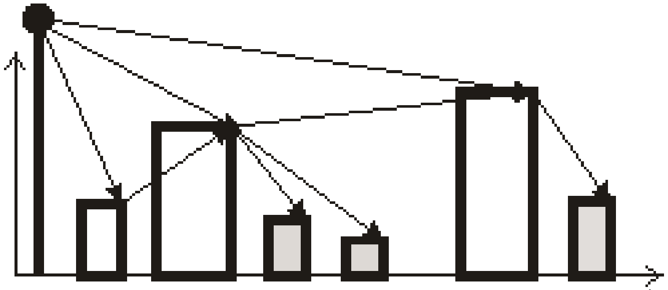
Feladat:

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:

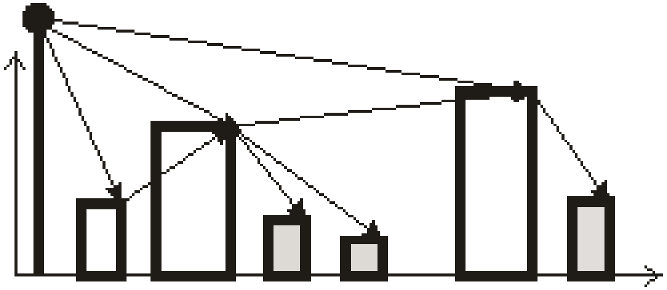
Az ábra szerint elég a házak bal felső sarkát ismerni!



Megoldás:

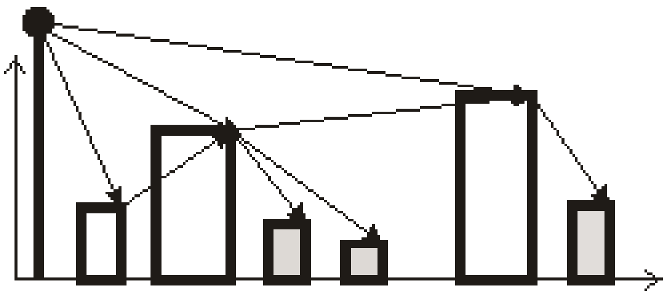
Legyen L a lámpa, H(i) az i-edik ház jobb felső sarkának helye, u pedig az utoljára megvilágított ház!

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



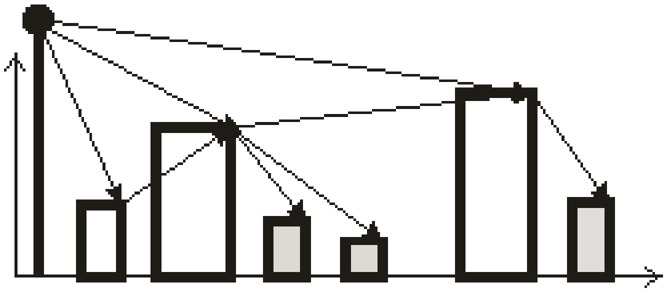
Geometriai algoritmusok

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



Változatok:

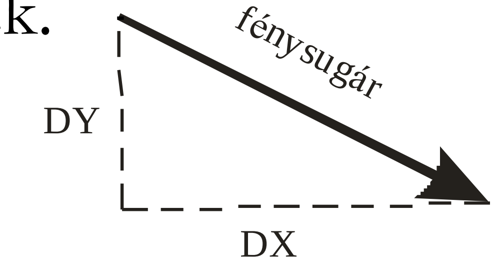
- Mi lenne, ha a lámpa a jobb szélen van?
- Mi lenne, ha a lámpa középen van?
- Mi lenne, ha a lámpa nem lenne magasabb minden épületnél?
- Mi lenne, ha több lámpa lenne?



Geometriai algoritmusok

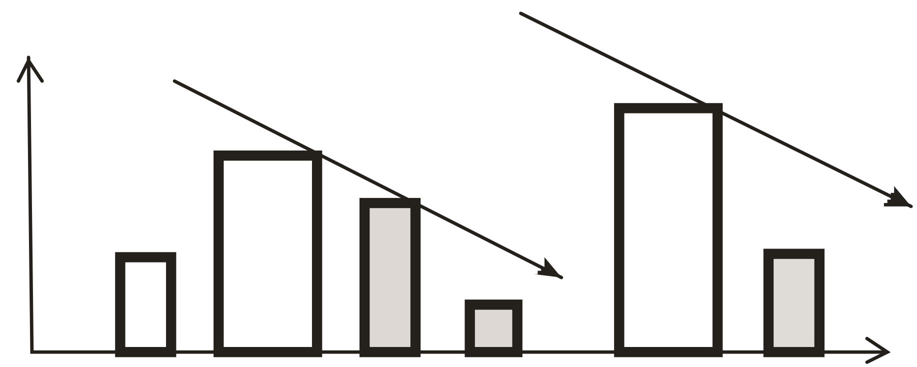
Feladat:

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?

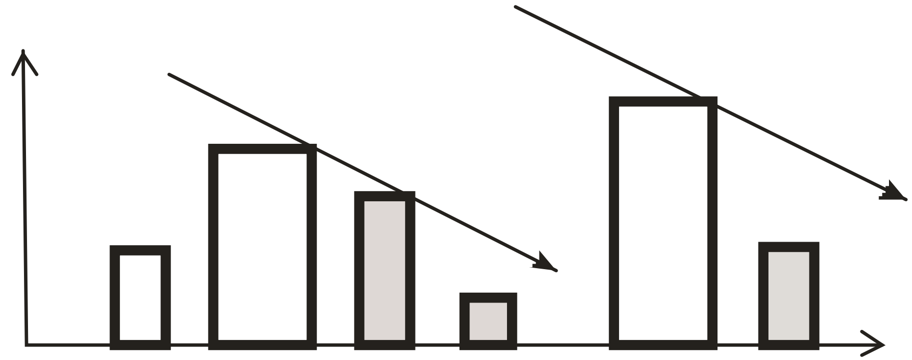
Az ábra szerint elég a házak bal felső sarkát ismerni!



Megoldás:

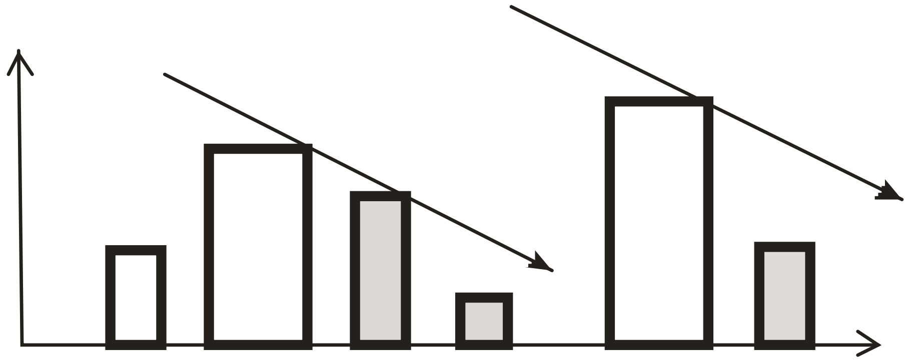
Legyen H(i) az i-edik ház jobb felső sarkának helye, u pedig az utoljára megvilágított ház!

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!



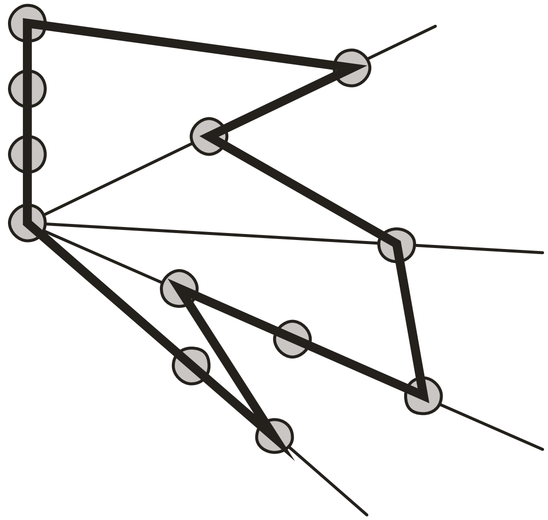
Geometriai algoritmusok

```
Világos(DX,DY,N,H,Db,Y):
  Db:=1; Y(Db):=1  {utolsó megvilágított}
  Ciklus i=2-től N-ig
    Ha fordul(H(Y(Db))-(DX,DY),H(Y(Db)),H(i))=-1
          akkor Db:=Db+1; Y(Db):=i
  Ciklus vége
Eljárás vége
```



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!



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.
```

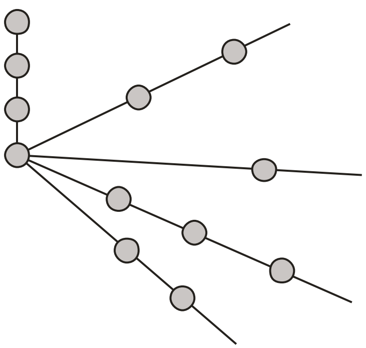
A rendezésnél persze elveszik a pontok régi sorszáma, azaz célszerű azt megőrizni, pl. a PONT típusba egy új mezőt, a sorszám mezőt felvéve.

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 a pontokat a sarokponton áthaladó, x-tengellyel párhuzamos egyenessel bezárt szög alapján, azonos szög esetén pedig a sarokponttól vett távolság szerint!



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_i 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 vagy a.x=b.x és a.y<b.y)
Függvény vége.
```

Geometriai algoritmusok

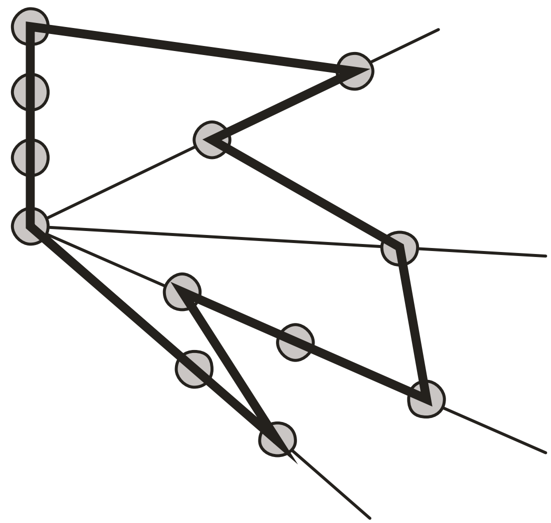
```
Rendez(N,P):
  Ciklus i=2-től N-1-ig
    min:=i
    Ciklus j=i+1-től N-ig
      Ha kisebb(P(1),P(j),P(i)) akkor min:=j
    Ciklus vége
    Csere(P(min),P(i))
  Ciklus vége
Eljárás vége.
```

Geometriai algoritmusok

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

Geometriai algoritmusok

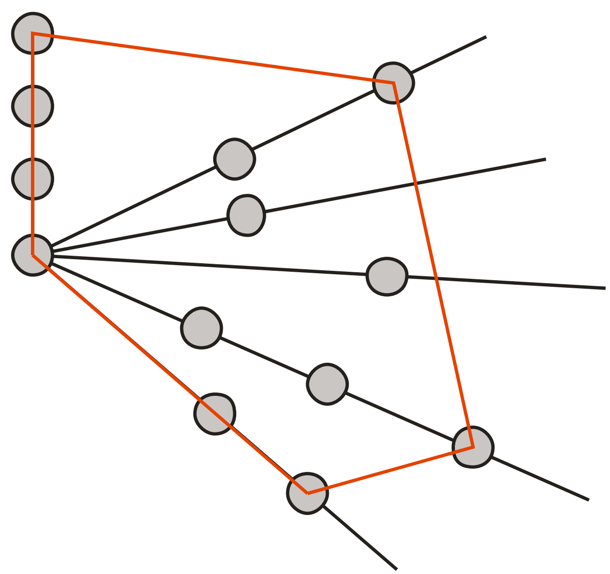
Kössük össze a pontokat a kapott sorrendben!



Geometriai algoritmusok

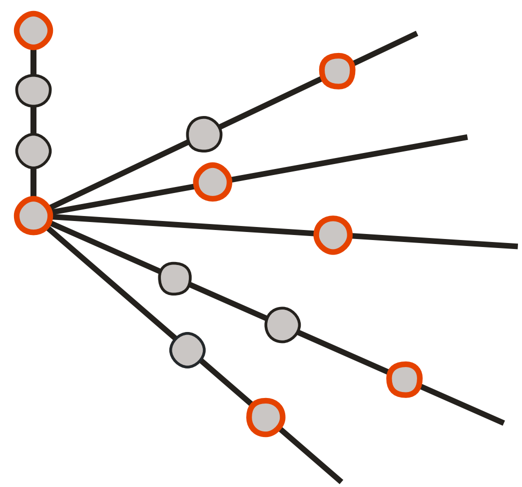
Feladat:

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



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.

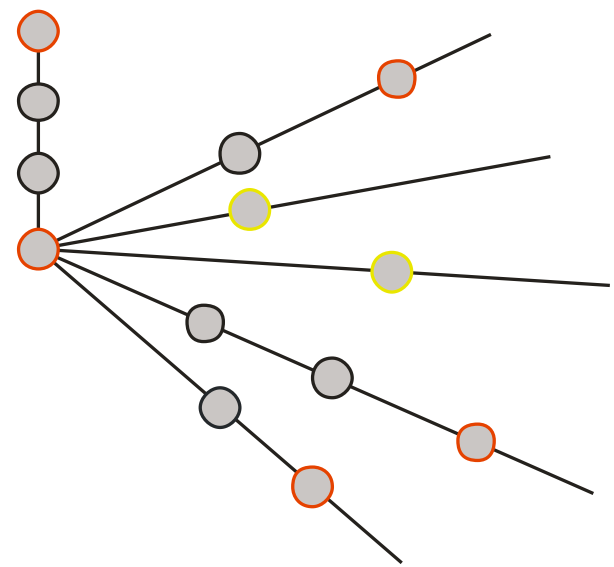


Geometriai algoritmusok

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!

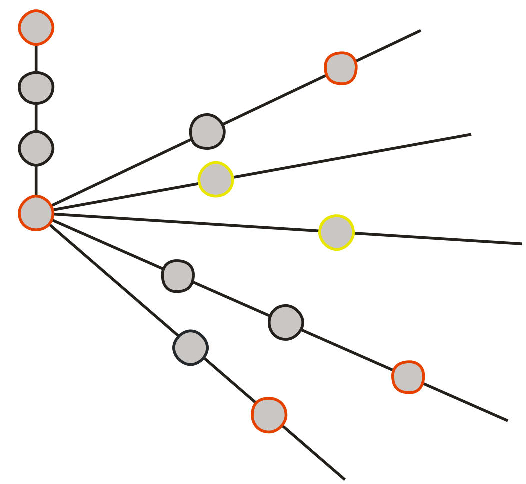
Ez az elv a korábban elhagyott pontokra is működik, azaz az elhagyás felesleges.



Geometriai algoritmusok

A rendezésnél persze elveszik a pontok régi sorszáma, azaz célszerű azt megőrizni, pl. a PONT típusba egy új mezőt, 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.
```

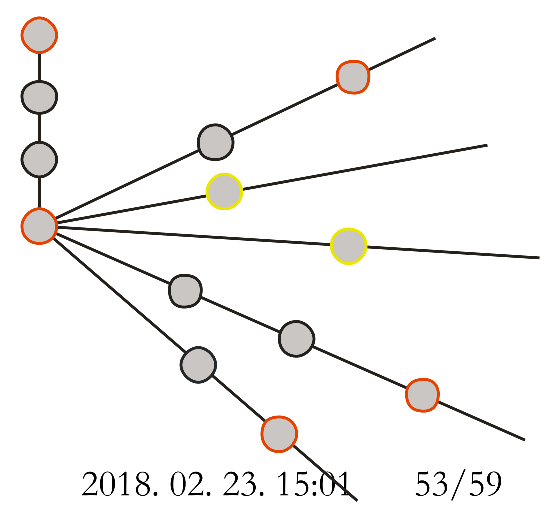


Geometriai algoritmusok

```
Körbejár(N,P,M,B):
  i:=3
  Ciklus amíg Fordul(P(1),P(i-1),P(i))=0
    i:=i+1
  Ciklus vége
  B(1):=1; B(2):=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 {B(M) nem jó}
       különben M:=M+1; B(M):=i; i:=i+1
  Ciklus vége
  M:=M-1
Eljárás vége.
```

Futási idő: N+K, ahol K≤N

(K=a B-ből kivett elemek száma)



Geometriai algoritmusok

Feladat:

Adott a síkon n darab pont, valamint egy további Q pont. A pontok (x,y) koordinátáikkal adottak. Adj meg egy P_i, P_j pontpárt úgy, hogy a Q pont a (P_i,P_j) szakaszon legyen!

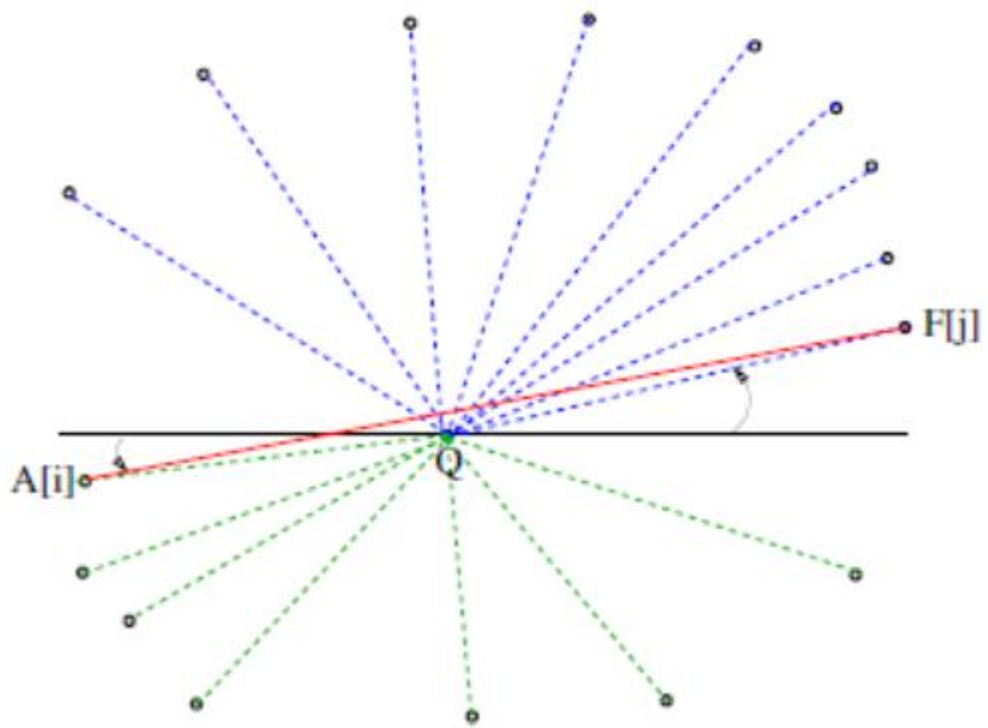
Megoldás

Osszuk két diszjunkt részhalmazba P pontjait aszerint, hogy a Q-n átmenő, x-tengellyel párhuzamos egyenes melyik oldalára esnek!

Rendezzük őket óramutató járásával ellentétes irányba!

Geometriai algoritmusok

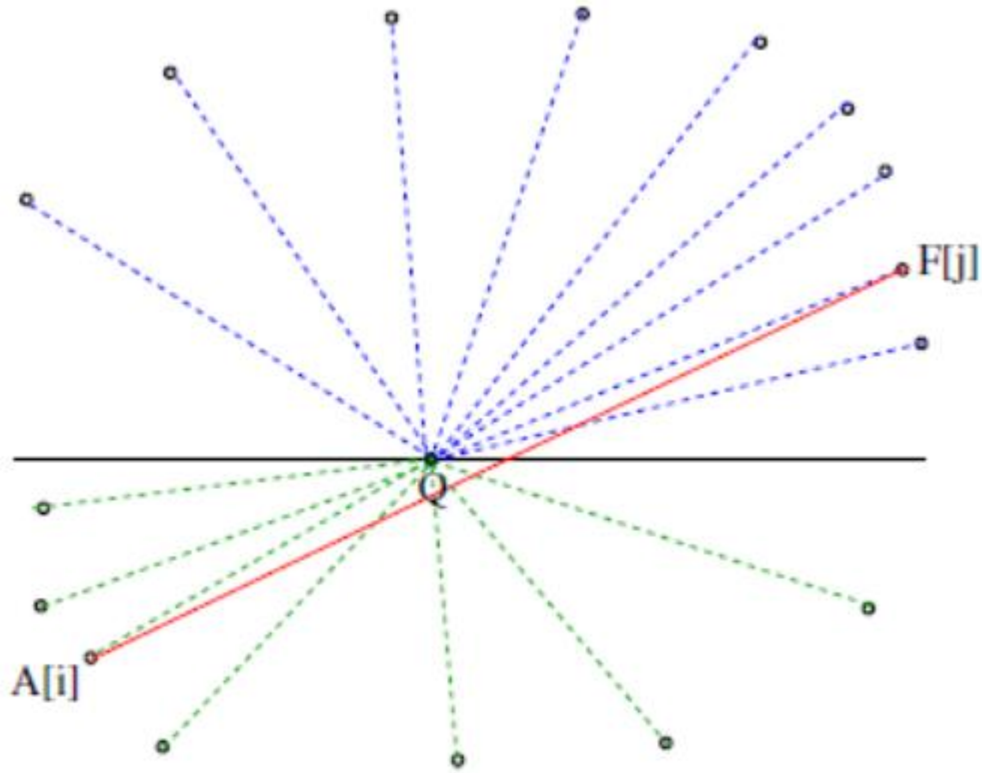
Az alábbi esetben az alsó pontoknál kell tovább-lépni, azaz i:=i+1!



Geometriai algoritmusok

Az alábbi esetben a felső pontoknál kell tovább-lépni, azaz j:=j+1!

Megállunk, ha a Q pont rajta van az (A[i],F[j]) egyenesen, vagy ha körbeértünk.



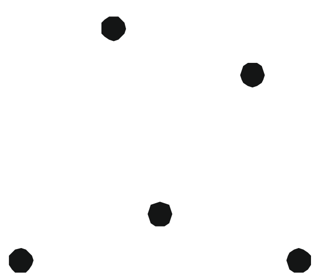
Geometriai algoritmusok

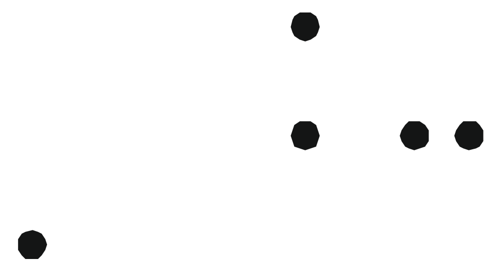
```
Keresés(N,P,Q,Van,i,j):
  Szétválogat(N,P,Q,NA,A,Aindex,NF,F,Findex)
  Rendez(NA,A,Aindex,Q); Rendez(NF,F,Findex,Q);
  i:=1; j:=1; van:=hamis
  Ciklus amíg i≤NA és j≤NF és nem van
    f:=Fordul(A[i],F[j],Q)
    Ha f<0 akkor j:=j+1
    különben ha f>0 akkor i:=i+1
    különben van:=igaz
  Ciklus vége
  Ha van akkor i:=Aindex(i); j:=Findex(j)
Eljárás vége.
```

Négyszög

A síkon négy pont által meghatározott négyszöget konvexnek nevezünk, ha mind a négy csúcsához tartozó szög kisebb 180 foknál.

Készíts programot, amely eldönti, hogy adott ponthalmazoknak van-e olyan négy pontja, amelyek konvex négyszöget alkotnak!





Geometriai algoritmusok
előadás vége