Belépő a tudás közösségébe

Informatika szakköri segédanyag

Geometriai algoritmusok

Bende Imre, Heizlerné Bakonyi Viktória, Menyhárt László, Szlávi Péter, Törley Gábor, Zsakó László

Szerkesztő: Abonyi-Tóth Andor, Zsakó László

A kiadvány „A felsőoktatásba bekerülést elősegítő készségfejlesztő és kommunikációs programok megvalósítása, valamint az MTMI szakok népszerűsítése a felsőoktatásban” (EFOP-3.4.4-16-2017-006) című pályázat keretében készült 2018-ban.

Eötvös Loránd Tudományegyetem
Informatikai Kar

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Áttekintés

Ebben az algoritmus típusban csak síkbeli geometriai feladatokkal foglalkozunk, azon belül is olyanokkal, ahol a pontok koordinátái egész számok, azaz nem kell lebegőpontos aritmetikával (és annak pontosságával) foglalkoznunk. A pontok szakaszokat alkothatnak, a szakaszok pedig sokszögek oldalai lehetnek.

A pontokat egy rekordban x- és y-koordinátával, valamint egy azonosító sorszámmal adjuk meg:

```
Pont=Rekord(x,y: Egész; sorszám: Egész)
```

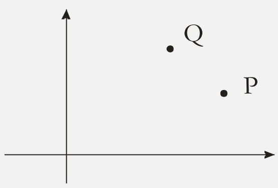
A feladattípus megoldását egy jól definiált függvénykönyvtárra építjük.

Geometriai alapfeladatok

Feladat

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

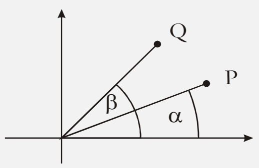
$$Irány(P,Q) = \begin{cases} -1, & ha \quad balra \\ +1, & ha \quad jobbra \\ 0, & ha \quad egy\ irányban \end{cases}$$

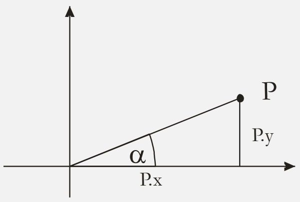


Értelmezés:

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

α<β → tan(α)<tan(β), ahol tan(α)=P.y/P.x, tan(β)=Q.y/Q.x





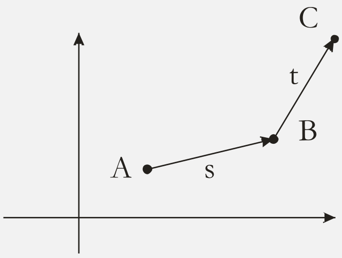
α<β ↔ tan(α)<tan(β) ↔ P.y/P.x<Q.y/Q.x ↔ P.y*Q.x<Q.y*P.x ↔P.y*Q.x–Q.y*P.x<0

```
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.
```

C++	cpp/1_Irany/feladat.cpp
C#	cs/1_Irany/feladat.cs
Java	java/1_Irany/feladat.java
Pascal	pas/1_Irany/feladat.pas
Python	py/1_Irany/feladat.py

Feladat

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



Megoldás

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)

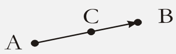
Ezzel ekvivalens feladat: Az (A,B)-n átmenő egyenestől a C pont balra van, vagy jobbra van, vagy az (A,B) egyenesen van?

```
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.
```

C++	cpp/2_Fordul/feladat.cpp
C#	cs/2_Fordul/feladat.cs
Java	java/2_Fordul/feladat.java
Pascal	pas/2_Fordul/feladat.pas
Python	py/2_Fordul/feladat.py

Feladat

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



Megoldás

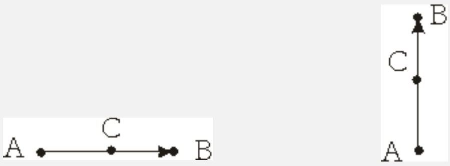
- Biztos nincs rajta, ha az A-B-C úton valamerre fordulni kell!
- Ha nem kell fordulni, akkor A és B között kell lennie!

```
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.
```

A fenti ábra szerint a C pont az A és a B pont között van, ha az A pont x-koordinátája a legkisebb, a C ponté van középen és a B pont x-koordinátája a legnagyobb. Az ábrán az A és a B pont felcserélhető, azaz lehet, hogy a B pont x-koordinátája a legkisebb és az A ponté a legnagyobb. Ugyanez az állítás igaz a pontok y-koordinátájára is.

```
Közte(r,s,t):
  Közte:=r≤s és s≤t vagy t≤s és s≤r
Függvény vége.
```

Sajnos az x- és az y-koordináta vizsgálatának választása nem alternatíva, vannak ugyanis speciális esetek, amikor csak az egyik ad helyes eredményt:

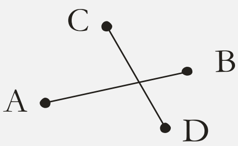


C++	cpp/3_Rajta/feladat.cpp
C#	cs/3_Rajta/feladat.cs
Java	java/3_Rajta/feladat.java
Pascal	pas/3_Rajta/feladat.pas
Python	py/3_Rajta/feladat.py

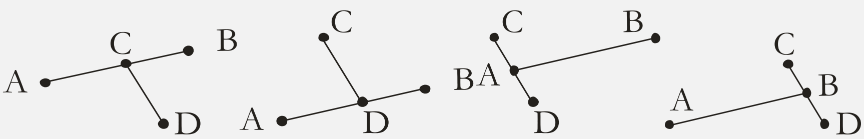
Feladat

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

Lehetséges esetek:



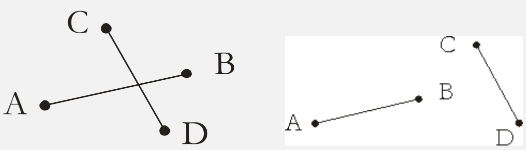
4 speciális esetet különböztethetünk meg, amelyek az előző feladatban definiált `Rajta` függvénynyel megoldhatóak:



Az általános esetben az A-B szakasz egyik oldalán van a C, másik oldalán a D; illetve a C-D szakasz egyik odalán van az A, és a másik oldalán a B.

```
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.
```

Nem elég az A-B-hez ellenkező irányban levő C és D pontot nézni, mert:



Minta kódok.

C++	cpp/4_Metszi/feladat.cpp
C#	cs/4_Metszi/feladat.cs
Java	java/4_Metszi/feladat.java
Pascal	pas/4_Metszi/feladat.pas
Python	py/4_Metszi/feladat.py

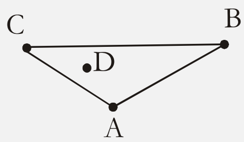
Elemi geometriai feladatok

Feladat

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

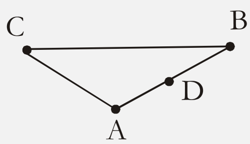
Megoldás

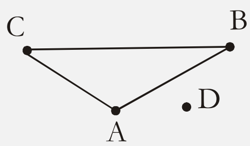
Belül van, ha a háromszöget A→B→C→A sorrendben körbejárva a D pont vagy mindig balra, vagy (ellenkező irányban bejárva) mindig jobbra van.




```
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.
```

Ha a D pont nem belső pont, akkor a három `Fordul` függvény nem adhat egyforma eredményt. Az alábbi ábrán a bal oldali háromszögnél az egyik `Fordul` értéke 0, a jobb oldali ábrán pedig van +1 és -0 értékű is:





Ha a határvonalat is belső pontnak számítjuk, akkor a megoldást kiegészítjük három `Rajta` függvény hívásával:

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

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

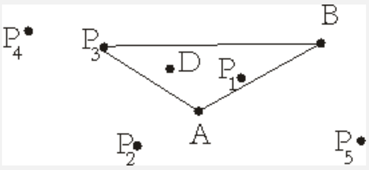
Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

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!

Megoldás

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



A megoldás egy keresés programozási tétel:

```
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.
```

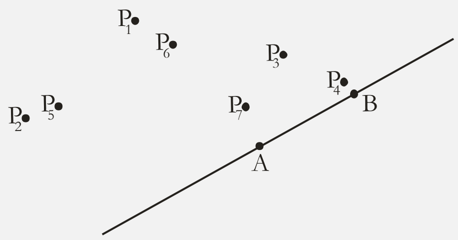
A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

Feladat

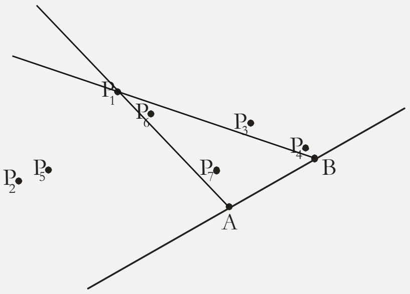
Adott A és B pont esetén adjunk meg további N pont közül egy P_i pontot úgy, hogy az (A,B,P_i) háromszög belsejében egyetlen más pont se legyen!

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



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



A potenciális jelölttel alkotott háromszögben talált újabb pont esetén az újabb pont lesz a potenciális jelölt, a vele (és az ere3deti két ponttal) alkotott háromszög ugyanis mindig része a korábbi jelölt által alkotott háromszögnek.

```
Kiválasztás(A,B,N,P,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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

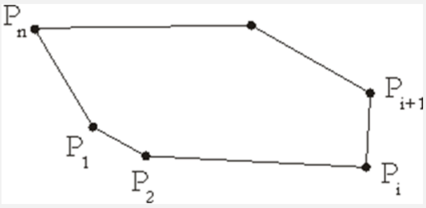
Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

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!



Egy eldöntés programozási tételt kell alkalmazni, akkor konvex a sokszög, ha minden i-re a P_i-P_{i+1}-P_{i+2} ponthármasnál balra kell fordulni. Az indexek ilyen megadása a sokszögvonal végén (az n. pontnál) problémás, de az egyszerűsítés kedvéért felveszünk két új pontot, $P_{n+1}=P_1$, $P_{n+2}=P_2$, így más minden egységesen kezelhető.

```
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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

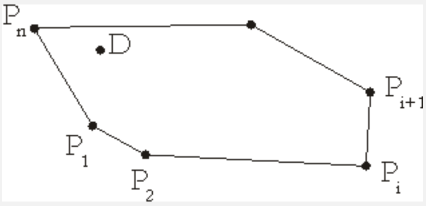
Feladat

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

Megoldás

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

Tehát ebben az esetben is egy eldöntés tételt kell alkalmaznunk.




```
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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

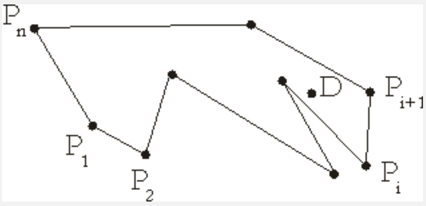
Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

Feladat

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

Probléma

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



Megoldás

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

A külső pont biztosan a sokszög alatt legyen, azaz az y-koordinátája legyen kisebb minden pont y-koordinátájánál!

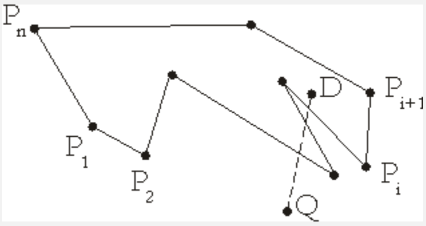
$$Q.y := \min_{i=1,\dots,N}(P_i.y)-1$$

Nem szerencsés, ha (D,Q) szakasz átmegy a sokszög valamely csúcsán (értelmezési kérdések merülhetnek fel), ezért úgy válasszuk a Q pont x-koordinátáját, hogy ez ne fordulhasson elő!

Ehhez vegyük a sokszög olyan pontjait, amelyek a D ponttól balra találhatóak (x-koordinátájuk kisebb a D pont x-koordinátájánál), s válasszuk közülük a legnagyobb x-koordinátát a Q pont x-koordinátájának!

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

Ha a (D,Q) szakasz a sokszög páratlan számú oldalát metszi, akkor a D pont belső pont, páros esetben pedig külső. Itt tehát egy megszámolás programozási tételt kell alkalmaznunk.



```
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!

```
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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

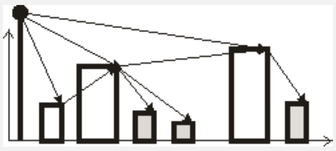
Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

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!

```
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.
```

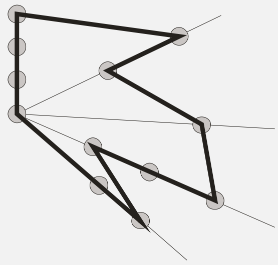
A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

Forgásirány szerinti rendezésen alapuló feladatok

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

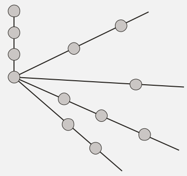
Válasszuk ki a legkisebb x-koordinátájú pontot, ha több ilyen van, akkor ezek közül válasszuk a legkisebb y-koordinátájút! Ezt nevezzük (bal-alsó) sarokpontnak és cseréljük meg az első ponttal!

```
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.
```

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!

```
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.



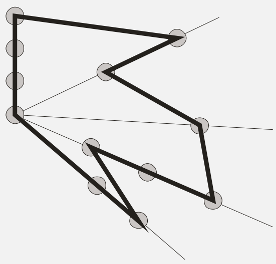
A sarokpont legyen az első, és p_i előbb legyen mint p_j akkor és csak akkor, ha a $p_1 \rightarrow p_i \rightarrow p_j$ úton balra kell fordulni, vagy nem kell fordulni, de p_i van közelebb a p_1-hez!

```
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.

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.
```

Kössük össze a pontokat a kapott sorrendben! Ez majdnem minden esetben jó, kivéve az utolsó iránynál, ahol kívülről befelé kell a pontokat összekötni, azaz a sorrendjüket meg kell fordítani.

```
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.
```



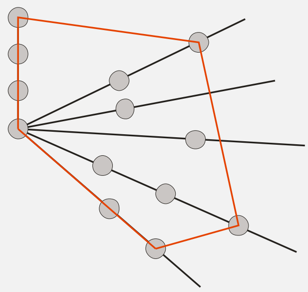
Az így kapott sokszögvonal sehol sem metszi önmagát, mert a sokszög oldalai vagy az első pontból kiinduló félegyeneseken vannak, vagy pedig két félegyenes egyetlen szakasszal kötnek össze.

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	24. Zárt poligon készítése

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!



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 csúcs-pontjai a konvex buroknak.

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.

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.
```

A konvex burok (B vektor) első két pontját biztosan ismerjük, ez a sarokpont, valamint az első irányban legmesszebb levő pont. A további pontoknál az utolsó két burokbeli ponthoz hasonlítjuk az i-ediket. Ha balra kell fordulni az utolsó burokbelinél, akkor az i-ediket felvesszük potenciális burokbeli pontnak. Ha nem balra kell fordulni, akkor az utolsó burokbeli pontot elvetjük.

```
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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	

Feladat

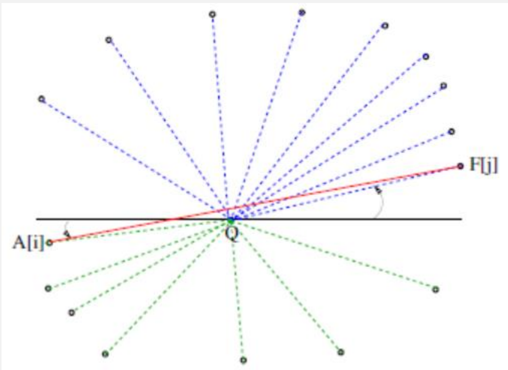
Adott a síkon n darab pont, valamint egy további Q pont. A pontok (x,y) koordinátáikkal adottak. Adj meg egy P_i, P_j pontpárt úgy, hogy a Q pont a (P_i,P_j) szakaszon legyen!

Megoldás

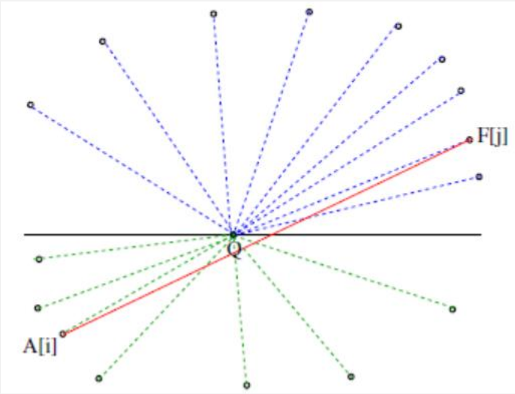
Osszuk két diszjunkt A és B részhalmazba P pontjait aszerint, hogy a Q-n átmenő, x-tengellyel párhuzamos egyenes melyik oldalára esnek! A Q-n átmenő vízszintes egyenesen levő pontok közül a Q-tól balra levők az A, a jobbra levők a B halmazba kerüljenek! (szétválogatás közben persze meg kel őriznünk az egyes pontok eredeti indexeit is.)

Rendezzük a két halmazt az óramutató járásával ellentétes irányba!

Az alábbi esetben az alsó pontoknál kell továbblépni, azaz i:=i+1!



Az alábbi esetben a felső pontoknál kell továbblépni, azaz j:=j+1!



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

```
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.
```

A feladat megoldása tesztelhető az elkészült forráskód feltöltésével itt:

Weboldal	https://mester.inf.elte.hu/
Szint	Haladó
Téma	Geometriai algoritmusok
Feladat	