Algoritmusok és adatszerkezetek II.

Geometriai algoritmusok

Szegedi Tudományegyetem

Definíció

A $P_3 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}$ pontot $P_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}$ és $P_2 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$ pontok **konvex kombináció**jának nevezzük, amennyiben $x_3 = (1-\alpha)x_1 + \alpha x_2$, valamint $y_3 = (1-\alpha)y_1 + \alpha y_2$ teljesül valamely $0 \leq \alpha \leq 1$-ra

Definíció

$\overline{P_1 P_2}$ szakasz a P_1 és P_2 pontokból konvex kombinációinak halmaza

Megjegyzés

Ha a pontok sorrendje is számít, irányított szakaszról beszélünk, és $\overrightarrow{P_1 P_2}$ módon jelöljük
$\vec{p}$-vel $\overrightarrow{OP}$-t, vagyis az O origóból a P-be menő irányított szakaszt (vektort) jelöljük

A keresztszorzat

$P_1 \times P_2$ keresztszorzata

$$\det\left(\begin{bmatrix} x_1 & x_2 \\ y_1 & y_2 \end{bmatrix}\right) = x_1 y_2 - x_2 y_1 = P_1 \times P_2 = -P_2 \times P_1$$

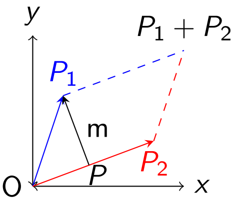
Megjegyzés

A keresztszorzat valójában háromdimenziós fogalom: egy $\overrightarrow{p_1}$-re és $\overrightarrow{p_2}$-re merőleges, velük jobbsodrású rendszert alkotó vektor, melynek hossza $|x_1 y_2 - x_2 y_1|$.

Forgásirány

Keresztszorzat mint előjeles terület

$P_1 \times P_2$ megadja az O, P_1, P_2, $P_1 + P_2$ koordinátákkal rendelkező paralelogramma előjeles területét



- $\mathbf{P_1 \times P_2 < 0 \Rightarrow P_1}$**-ből jobbra fordulva érjük el** $\mathbf{P_2}$**-t**
- $P_1 \times P_2 > 0 \Rightarrow P_1$-ből balra fordulva érjük el P_2-t
- $P_1 \times P_2 = 0 \Rightarrow P_1$ és P_2 kollineáris

Merre fordul a következő szakasz?

- $\overline{P_0 P_1}$ és $\overline{P_1 P_2}$ szakaszokat folyamatosan bejárva merre kell fordulni P_1 pontban?
- Az előzőekben lényegében az origó viselkedett P_0-ként

Merre fordul a következő szakasz?

- $\overline{P_0P_1}$ és $\overline{P_1P_2}$ szakaszokat folyamatosan bejárva merre kell fordulni P_1 pontban?
- Az előzőekben lényegében az origó viselkedett P_0-ként

Ötlet: tegyünk úgy, mintha P_0 lenne az origó

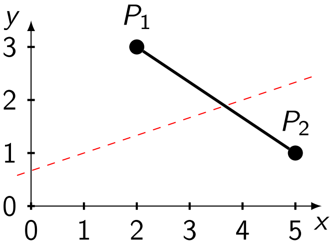
$$(P_1 - P_0) \times (P_2 - P_0) = \det\left(\begin{bmatrix} x_1 - x_0 & x_2 - x_0 \\ y_1 - y_0 & y_2 - y_0 \end{bmatrix}\right)$$

- Szemléletesen: P_1-ből és P_2-ből P_0-t kivonva P_0 központúvá tesszük a koordinátarendszerünket

Szakasz átfogása

Átfogó szakasz

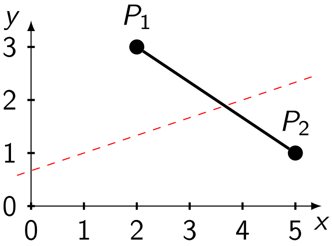
Egy $\overline{P_1 P_2}$ szakasz átfog egy egyenest, ha a P_1 pont az egyenes egyik oldalára, P_2 pont pedig a másik oldalára esik



Szakasz átfogása

Átfogó szakasz

Egy $\overline{P_1 P_2}$ szakasz átfog egy egyenest, ha a P_1 pont az egyenes egyik oldalára, P_2 pont pedig a másik oldalára esik



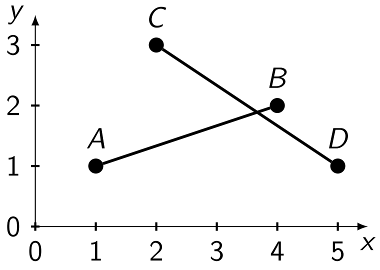
Átfedés meglétének eldöntése

Egy (kevéssé hatékony) lehetőség, ha az egyenes egyenletét kiszámolva döntünk P_1 és P_2 relatív helyzetéről
Támaszkodjunk helyette a forgásirányokra!

Egymást metsző szakaszok

Szükségesség

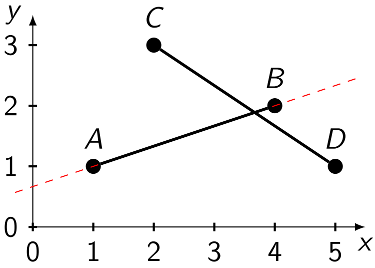
$\overline{CD}$ úgy metszheti $\overline{AB}$ szakaszt, ha $\overline{CD}$ átfogja az $\overline{AB}$ szakaszra illeszkedő egyenest.

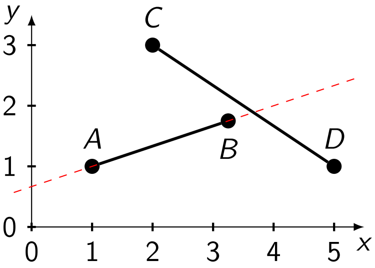


Egymást metsző szakaszok

Szükségesség

$\overline{CD}$ úgy metszheti $\overline{AB}$ szakaszt, ha $\overline{CD}$ átfogja az $\overline{AB}$ szakaszra illeszkedő egyenest.





Metszés vizsgálata

```
ForgásIrány(X, Y, Z) {
   return (Y-X)×(Z-X)
}

MetszőSzakaszok(A, B, C, D) {
   d1 =  ForgásIrány(A, B, C)
   d2 =  ForgásIrány(A, B, D)
   d3 =  ForgásIrány(C, D, A)
   d4 =  ForgásIrány(C, D, B)
   return d1 * d2 < 0 és d3*d4 < 0
}
```

Metszés vizsgálata

```
FORGÁSIRÁNY(X, Y, Z) {
  return (Y-X)×(Z-X)
}

METSZŐSZAKASZOK(A, B, C, D) {
  d1 = FORGÁSIRÁNY(A, B, C)
  d2 = FORGÁSIRÁNY(A, B, D)
  d3 = FORGÁSIRÁNY(C, D, A)
  d4 = FORGÁSIRÁNY(C, D, B)
  return d1 * d2 < 0 és d3*d4 < 0
}
```

Ezzel csak „valódi” metszéseket találunk meg, a szakaszra illeszkedő végpontú szakaszt nem kezeltük így

Metsző szakaszpár keresése

- Adott szakaszok n elemű halmaza, és tudni szeretnénk, hogy van-e köztük egymást metsző szakaszpár
- Nyers erővel $\binom{n}{2} = O(n^2)$
- Bizonyos egyszerűsítő feltételezések mellett $O(n \log(n))$ is megoldható
  - y tengellyel párhuzamos szakaszokat nem kezelünk
  - Adott szakaszok között nincs 3 egy pontban metsző

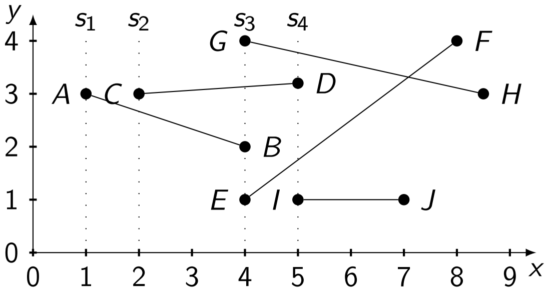
Metsző szakaszpár keresése – söprés

Söprés

Söprés során egy képzeletbeli függőleges *söprő egyenes* halad át geometriai objektumok halmazán (általában balról jobbra)

Két szakasz összehasonlítása adott x koordináta mentén

s_1 szakasz fölötte van s_2-nek x-nél ($s_1 \succ_x s_2$), ha s_1 y-koordinátája nagyobb s_2 y-koordinátájánál adott x-koordináta mentén.

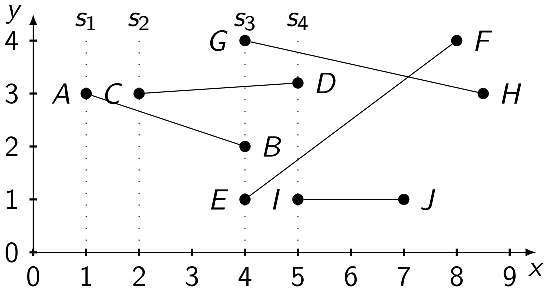


Söprés

Söprés során egy képzeletbeli függőleges *söprő egyenes* halad át geometriai objektumok halmazán (általában balról jobbra)

Két szakasz összehasonlítása adott x koordináta mentén

s_1 szakasz fölötte van s_2-nek x-nél ($s_1 \succ_x s_2$), ha s_1 y-koordinátája nagyobb s_2 y-koordinátájánál adott x-koordináta mentén.



Példák

$\overline{GH} \succ_4 \overline{EF}$, de $\overline{EF} \succ_8 \overline{GH}$

Szakaszpár metszése a söprő egyenes szemszögéből

- Bármely adott x értékre a $\succ_x$ reláció az x-nél lévő söprő egyenest metsző szakaszok teljes rendezése

Kulcsészrevétel

Ha $\exists$ egymást metsző szakaszpár $\Rightarrow$ $\exists$ söprő egyenes, mely mentén való rendezés esetén azok egymás után következnek

- A söprő egyenes mozgatása
  - Elég a szakaszvégpontokban a be,-és kilépő szakaszok alapján összehasonlításokat végezni
- Kétféle adathalmazt kell kezelni a keresés során
  - Söprő egyenes állapotleírása
  - Esetpontok rendezett listája

Szakaszpár metszése – állapotleírás és esetpontok

- A söprő egyenes állapotleírása a szakaszok adott egyenes menti $\succ$ teljes rendezési reláció szerinti rendezését tartalmazza
- A söprő egyenes állapotleírásában változás csak esetpontokban (=szakaszvégpontokban) történik
- Esetpontok rendezése
  - Kovertikális (azonos x-koordinátájú) szakaszvégpontok esetén a bal/belépő végpontokat a jobb/kilépő végpontok elé soroljuk
- Elegendő azt vizsgálni csupán, hogy a
  - belépő szakaszok metszik-e megelőzőjüket/rákövetkezőjüket
  - kilépő szakaszok megelőzője és rákövetkezője metszi-e egymást

Szakaszpár metszése – állapotleírás és esetpontok

- A söprő egyenes állapotleírása a szakaszok adott egyenes menti $\succ$ teljes rendezési reláció szerinti rendezését tartalmazza
- A söprő egyenes állapotleírásában változás csak esetpontokban (=szakaszvégpontokban) történik
- Esetpontok rendezése
  - Kovertikális (azonos x-koordinátájú) szakaszvégpontok esetén a bal/belépő végpontokat a jobb/kilépő végpontok elé soroljuk
- Elegendő azt vizsgálni csupán, hogy a
  - belépő szakaszok metszik-e megelőzőjüket/rákövetkezőjüket
  - kilépő szakaszok megelőzője és rákövetkezője metszi-e egymást

Fontos

A mindenkori állapotleírást kiegyensúlyozott keresőfában tároljuk

Metsző szakaszpár keresése

```
VAN-E-METSZŐ-SZAKASZPÁR(S) { // T az állapotleírás fája
  L = S-beli szakaszvégpontok rendezett listája
  for p in L
  do
    if p egy s szakasz bal végpontja {
      BESZÚR(T,s)
      if MEGELŐZ(T,s) vagy RÁKÖVET(T,s) metszi s-et {
        return IGAZ
      }
    }
    if p egy s szakasz jobb végpontja {
      if MEGELŐZ(T,s) metszi RÁKÖVET(T,s)-t {
        return IGAZ
      }
      TÖRÖL(T,s)
  return HAMIS
}
```

Metsző szakaszpár keresésének futási ideje

- Állapotleírás T kiegyensúlyozott fájának létrehozása $O(1)$
- Szakaszvégpontok rendezése $O(n \log n)$
- `for` ciklus "hossza" legfeljebb $2n = O(n)$
- Megelőz, illetve Rákövet metódusok végrehajtása $O(\log n)$

Metsző szakaszpár keresésének futási ideje

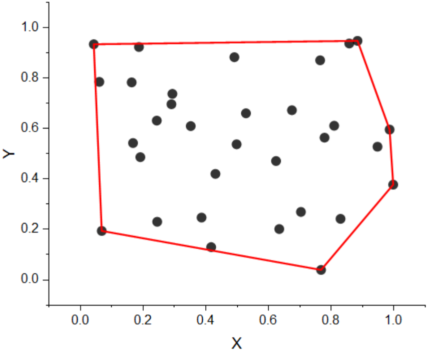
- Állapotleírás T kiegyensúlyozott fájának létrehozása $O(1)$
- Szakaszvégpontok rendezése $O(n \log n)$
- `for` ciklus "hossza" legfeljebb $2n = O(n)$
- Megelőz, illetve Rákövet metódusok végrehajtása $O(\log n)$

$\Rightarrow$ Összességében $O(n \log n)$

Konvex burok definíciója

Konvex burok

Q ponthalmaz konvex burka az a legkisebb P konvex poligon, amelyre Q minden pontja vagy P határán van, vagy a belsejében.



Q konvex burkát CH(Q)-val jelöljük. $CH(Q)$-ra természetesen igaz, hogy $CH(Q) \subseteq Q$.

Konvex burok meghatározására szolgáló algoritmusok

- Különféle megközelítések léteznek
  1. **Forgatásos söprés** (Graham-féle pásztázás is ilyen)
  2. Növekményes módszer: ''balról jobbra'' számítjuk ki a CH-t
  3. Oszd-meg-és-uralkodj: CH számítása a pontok részhalmazára, majd ezek egyesítése
  4. Eltávolító és kereső módszer
- n pont esetében többnyire $O(n \log n)$ futási idejű algoritmusok, de vannak $O(nh)$, illetve $O(n \log h)$ algoritmusok is[1]

[1] h a CH-ban lévő csúcsok száma

Konvex burok meghatározására szolgáló algoritmusok

- Különféle megközelítések léteznek
  1. **Forgatásos söprés** (Graham-féle pásztázás is ilyen)
  2. Növekményes módszer: ''balról jobbra'' számítjuk ki a CH-t
  3. Oszd-meg-és-uralkodj: CH számítása a pontok részhalmazára, majd ezek egyesítése
  4. Eltávolító és kereső módszer
- n pont esetében többnyire $O(n \log n)$ futási idejű algoritmusok, de vannak $O(nh)$, illetve $O(n \log h)$ algoritmusok is[1]

Megjegyzés

Egy $O(nh)$ algoritmusnak nyilván csak $h < \log n$ esetén van haszna

[1] h a CH-ban lévő csúcsok száma

Konvex burok – Graham-féle pásztázás

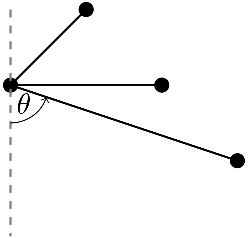
```
GRAHAM-PÁSZTÁZÁS(S) {
  P0 = minimális x-koordinátájú Q-beli pont (több ilyen
  esetén válasszuk az x-koordináta szerint is minimálisat)
  P = POLÁRSZÖGSZERINTRENDEZ(Q)
  S = VERMETLÉTESÍT()
  VEREMBE(P0, S)
  VEREMBE(P1, S)
  VEREMBE(P2, S)
  for i=3 to m {
    while LEGFELSŐ-ALATTI(S), LEGFELSŐ(S) és Pi
          pontok szöge nem fordul balra {
      VEREMBŐL(S)
    }
    VEREMBE(Pi, S)
  }
  return S
}
```

Polárszög szerinti rendezés

Pontok helyzetének polárkoordinátákkal történő megadása

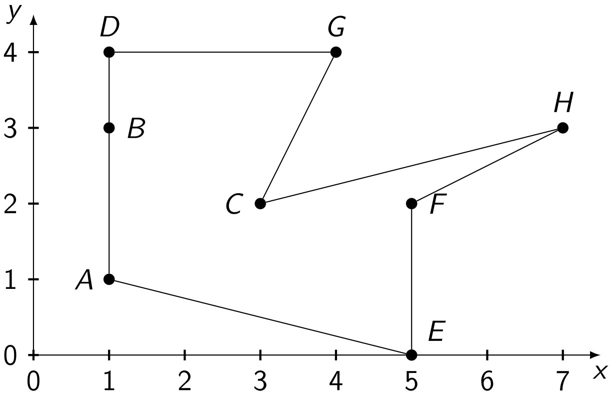
P pontot (x, y) koordinátapár helyett egy referenciaponttól vett távolság és egy referenciairánnyal bezárt szög párosaként adjuk meg

- A refereciaponttól számított $\frac{\Delta y}{\Delta x}$ eltérések szerinti sorrend adja a pontok polárszög szerinti rendezését
  - Azonos hányadossal rendelkező pontok közül azt soroljuk előbbre, amelyik a referenciaponthoz közelebb található[2]



[2]Kivéve, ha $\Delta x = 0$, mert akkor a másodlagos rangsorolást fordítva végezzük

- A pontokat a polárszöges rendezés sorrendjében összekötve megkapjuk a pontok által alkotott zárt, nem metsző poligont



A rendezés

1. A
2. E
3. F
4. H
5. C
6. G
7. D
8. B

Legtávolabbi pontpár megtalálása

- n elemű ponthalmazban találjuk meg azon (P_i, P_j) pontpárt, melyek a legtávolabb fekszenek egymástól
  - P_i és P_j pontok távolságát euklideszi értelemben véve $d(P_i, P_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$
- Nyers erővel ez is $\binom{n}{2} = O(n^2)$ összehasonlítás lenne

Észrevétel

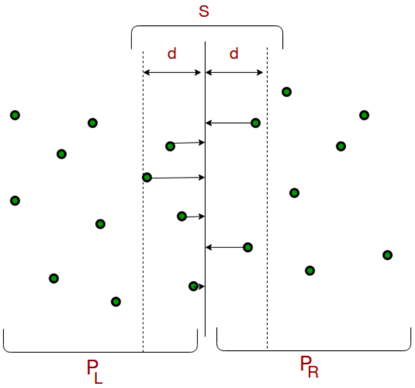
A ponthalmaz legtávolabbi pontpárja a CH-on található csúcspárok valamelyike kell legyen

Legközelebbi pontpár megtalálása

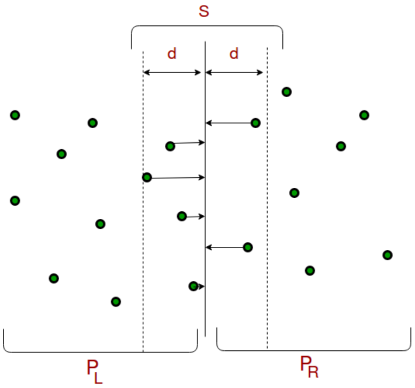
- Adott n elemű P ponthalmazra mi az a $(P_i, P_j) \in P \times P$ pontpár, ami a legközelebb helyezkedik el egymáshoz?
- Nyers erővel szintén $\binom{n}{2} = O(n^2)$
- Oszd meg és uralkodj eljárással $O(n \log(n))$ is megoldható

- Ha legfeljebb 3 pont maradt, akkor használjuk a nyers erő módszerét
- Egyébként hajtsuk végre a következőket
  - Vegyük azt az l egyenest, ami két egyenlő részre vágja a pontokat (P_L és P_R)
  - A P_L és P_R-beli pontok közül rekurzívan keressük meg a legközelebbi pontpárt $d = \min(d_L, d_R)$
  - Döntsük el, hogy találni-e olyan (P_i, P_j) pontpárt, melyre $P_i \in P_L$ és $P_j \in P_R$, továbbá távolságuk $d' < d$

Legközelebbi pontpár megtalálása – illusztráció



Legközelebbi pontpár megtalálása – illusztráció



Jó hír

Elegendő az y koordináta alapján vett rendezés szerinti 7 rákövetkező ponttal összevetni az S-beli pontokat