Vaje iz računske geometrije

Sergio Cabello

9. november 2010

naslov: Vaje iz računske geometrije

avtorske pravice: Sergio Cabello

izdaja: prva izdaja

založnik: samozaložba

avtor: Sergio Cabello

leto izida: 2010 (v Ljubljani)

natis: elektronsko gradivo
http://www.fmf.uni-lj.si/~cabello/gradiva/vajeracgeom.pdf

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

514.17(075.8)(076.1)
519.852(075.8)(076.1)

CABELLO, Sergio
Vaje iz računske geometrije [Elektronski vir] / Sergio Cabello. - 1. izd. - El. knjiga. - V Ljubljani : samozal., 2010

Način dostopa (URL): http://www.fmf.uni-lj.si/~cabello/gradiva/vajeracgeom.pdf

ISBN 978-961-276-043-4

253228800

Kazalo

Naloge so grobo organizirane po temah, vendar je klasifikacija včasih težka. V vsaki nalogi lahko predpostaviš smiseln *splošen položaj*, ki pa ga moraš natančno opisati. Ko opišeš algoritem, moraš vedno pokazati pravilnost in nadalje opisati in utemeljti časovno zahtevnost algoritma. Časovna zahtevnost se lahko nanaša na najslabši primer (*worst-case analysis*) ali na pričakovani primer (*average-case analysis*). Dobro je, da vsaj za nekatere vaje napišeš psevdokodo.

Za študiranje priporočam knjigo

> M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf: Computational Geometry, Algorithms and Applications. Springer.

Zahvala. Zahvaljujem se Mitji Trampušu za številne popravke. Seveda pa je avtor odgovoren za vse preostale napake.

1 Osnovne naloge

1.1. Naj bo $T(1) = O(1)$. Dokaži:

(a) Če je $T(n) = O(n) + 2T(n/2)$, potem je $T(n) = O(n \log n)$.

(b) Če je $T(n) = O(n) + T(n/2)$, potem je $T(n) = O(n)$.

1.2. Opiši algoritem, ki dobi seznam točk $p_1, p_2, \ldots, p_n$ v $\mathbb{R}^2$ in zbriše ponovljene točke, torej vrne množico $\{p_1, p_2, \ldots, p_n\}$. Algoritem mora imeti časovno zahtevnost $O(n \log n)$.

Opiši algoritem za isti problem v $\mathbb{R}^d$. Koliko časa potrebuje?

1.3. Podanih je n daljic v ravnini. Vemo, da so daljice robovi večkotnika, vendar niso nujno naštetí v "pravem" vrstnem redu. Opiši algoritem, ki rekonstruira večkotnik kot seznam naštetih oglišč v času $O(n \log n)$.

1.4. Za 3 točke $p = (p_x, p_y), q = (q_x, q_y), r = (r_x, r_y)$ v ravnini definiramo

$$D(p, q, r) = \begin{vmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{vmatrix}.$$

Dokaži, da je $D(p, q, r) > 0$ natanko tedaj, ko je zaporedje p, q, r zasuk na levo. Dokaži, da je $D(p, q, r) = 0$ natanko tedaj, ko so p, q, r kolinearne. Dokaži, da je $|D(p, q, r)|$ dvakrat ploščina trikotnika $\triangle pqr$.

1.5. Opiši algoritem, ki izračuna presečišča med podano premico in podano krožnico.

1.6. Naj bo P množica n točk v ravnini. Predpostavimo naslednji splošni položaj: nobene 3 točke niso kolinearne. Opiši algoritem, ki skonstruira v času $O(n \log n)$ večkotnik, ki ima množico P kot množico oglišč.

Pokaži primer množice točk P z naslednjo lastnostjo: obstaja eksponentno število večkotnikov, ki imajo množico P kot množico oglišč. To je, obstajati mora najmanj $1.0000001^{|P|}$ takih večkotnikov.

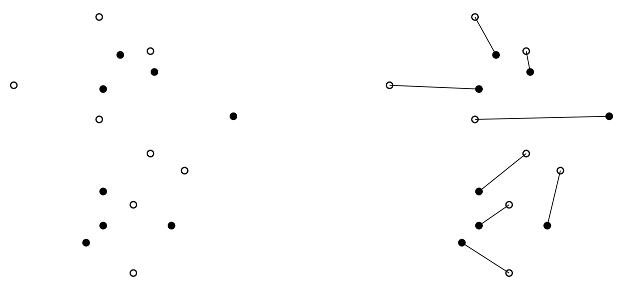
1.7. Podani sta enako veliki množici točk R (rdeč) in B (modri) v ravnini. Predpostavimo, da je $R \cap B = \emptyset$ in da niso nobene 3 točke kolinearne. Prirejanje $M \subset R \times B$ med R in B je *disjunktno prirejanje*, če so daljice $\overline{rb}$, $(r, b) \in M$, paroma disjunktne. Glej sliko 1.1.

(a) Naj bo M^* prirejanje med R in B, ki je najkrajše (vsota dolžin daljic). Dokaži, da je M^* disjunktno prirejanje.

(b) Ali lahko uporabljamo (a) za algoritem, ki vrne disjunktno prirejanje.

1.8. Podana sta množica točk P moči n in premica ℓ v ravnini. Zanima nas najmanjši krog D_{min}, ki vsebuje P in ima svoje središče na premici ℓ. Opiši algoritem, ki najde D_{min} v polinomskem času. (Čas $O(n \log n)$ je dosegljiv, ampak pretežek na začetku predmeta.)

1.9. Podana je množica n točk P v ravnini. Zanima nas, ali obstajata 2 unitarna kvadrata (obseg je 4), ki vsebujeta P in imata robove vzporedne z x-osjo ali z y-osjo.



Slika 1.1: Zgled za vajo 1.7.

Opiši algoritem, ki rešuje ta problem v polinomskem času. Kakšne podatkovne strukture bi bile koristne za bolj učinkovit algoritem?

1.10. Naj bo P množica n točk v ravniti, in naj bo L množica premic, ki grejo skozi 2 točki iz P. Opiši algoritem, ki vrne v času $o(n^2)$ premico iz L z največjim naklonom.

1.11. Naj bo D krog polmera 1 v ravnini in naj bo P množica n točk v D.

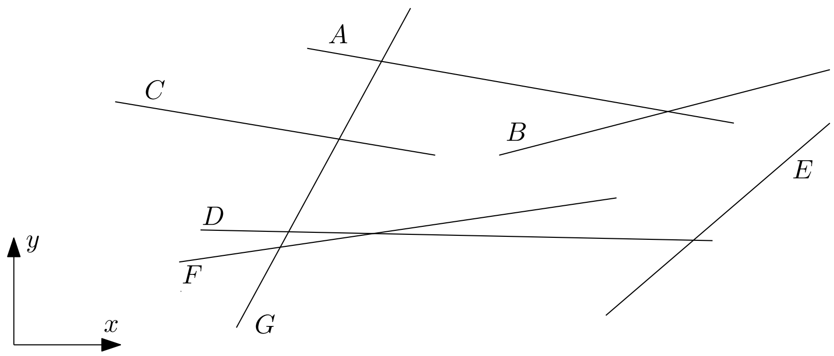
(a) Predpostavimo, da za vsak par $p, q \in P$ velja $|pq| \geq \delta$, kjer je $\delta > 0$ parameter. Dokaži zgornjo mejo za $|P|$, ki je tesna z O-notacijo. (Namig: uporabi argument s ploščino krogov.)

(b) Predpostavimo, da za vsako točko $p \in P$ velja $|\{q \in P \mid |pq| \leq \delta\}| \leq 20$. Dokaži tesno zgornjo mejo za $|P|$. (Namig: ali lahko uporabimo (a) 20 krat?)

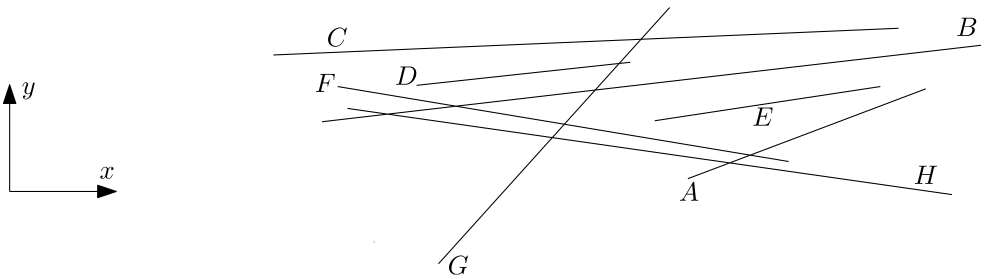
2 Algoritmi pometanja

Za algoritme pometanja opiši vedno glavno idejo, strukturo stanja, dogodke (ter kako jih algoritem obravnava) in invarianto zanke.

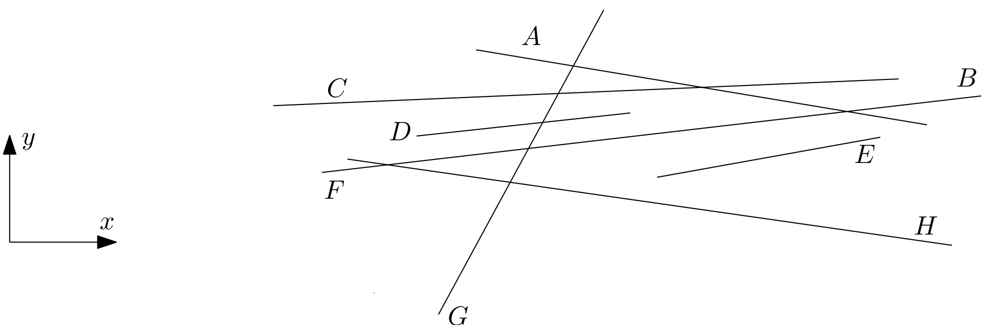
2.1. Na slikah 2.2-2.4 so podane množice daljic S. Njihova presečišča bi radi poiskali z uporabo algoritma pometanja, ki si ga spoznal na predavanjih. Navedi seznam parov daljic $\{s, s'\}$, $s, s' \in S$, za katere algoritem preveri, ali je $s \cap s'$ prazen ali ne. Pari morajo v seznamu nastopati v istem vrstnem redu, kot jih algoritem preveri. Če algoritem isti par daljic preveri večkrat, potem naj natančno tolikokrat nastopa tudi v seznamu.



Slika 2.2: Vaja 2.1.



Slika 2.3: Vaja 2.1.



Slika 2.4: Vaja 2.1.

2.2. Naj bo $S = \{s_1, s_2, \ldots, s_6\}$ množica daljic v ravnini. Daljice imajo naslednja krajišča: $s_1 = \overline{(0,0)(20,10)}$, $s_2 = \overline{(9,19)(9,6)}$, $s_3 = \overline{(2,7)(14,4)}$, $s_4 = \overline{(3,2)(15,2)}$, $s_5 = \overline{(5,4)(8,10)}$, $s_6 = \overline{(1,10)(16,5)}$. Za iskanje presečišč iz S uporabimo algoritem pometanja, ki ga smo opisali na predavanjih.

(a) Koliko dogodkov je?

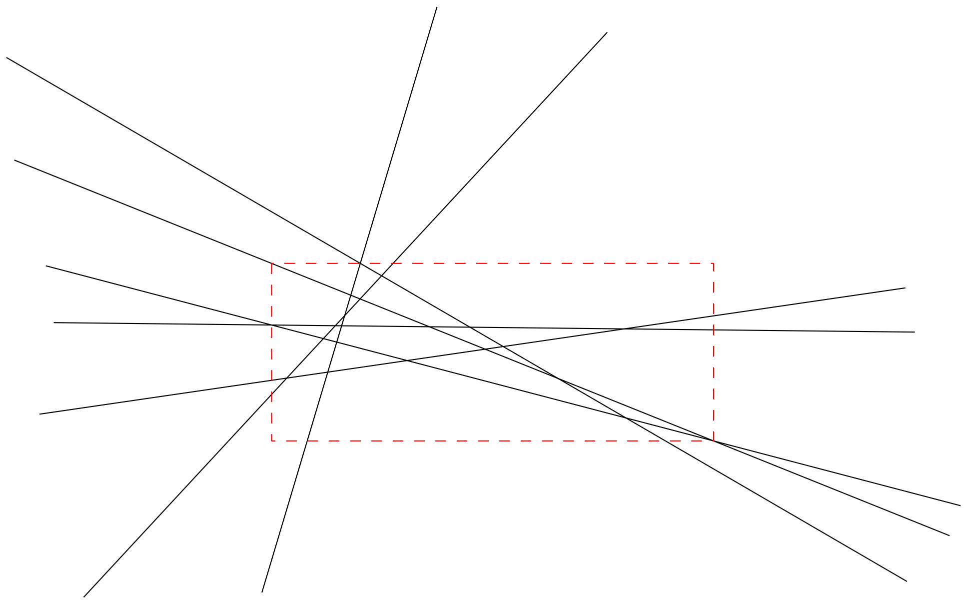
(b) Na katerih dogodkih najdemo nova presečišča, ki bodo zdaj dogodki?

(c) Opiši strukturo stanja po dogodku $(0,0)$ in pred dogodkom $(8,10)$.

2.3. Podana je množica paroma disjunktnih daljic S moči n. Vsaka daljica iz S ima eno skrajno točko na premici $y = 0$ in drugo skrajno točko na $y = 1$. Daljice v S razdelijo prostor med $y = 0$ in $y = 1$ na kose. Podrobno opiši (psevdokoda) podatkovno strukturo, ki jo lahko konstruiramo v času $O(n \log n)$ in odgovori v času $O(\log n)$ na poizvedbe naslednjega tipa: katere daljice definirajo kos, v katerem je poizvedbalna točka $p(p_x, p_y)$, $0 < p_y < 1$.

2.4. Na predavanju smo spoznali algoritem pometanja, ki najde v času $O((n + k) \log n)$ pare daljic, ki se sekajo, kjer je k število vrnjenih parov. Algoritem, ki ga smo opisali, porabi za delovanje $O(n + k)$ prostora. Opiši spremembe v algoritmu, s katerimi bi algoritem porabil samo $O(n)$ prostora. Torej algoritem vrne presečišča takoj, ko jih najde, in porabi samo $O(n)$ dodatnega prostora.

2.5. Podana je množica premic L moči n v ravnini. Naj bo I množica presečišč premic iz L. Opiši algoritem, ki v času $O(n \log n)$ vrne najmanjši pravokotnik, ki vsebuje I in ima robove vzporedne koordinatnim osem. Glej sliko 2.5.



Slika 2.5: Zgled za vajo 2.5.

2.6. Naj bo S množica n krožnic v ravnini. (Krožnica je krivulja, in sicer meja kroga.) Opiši output-sensitive algoritem pometanja, ki vrne presečišča med pari krožnic v času $O((n+k)\log n)$, kjer je k število vrnjenih presečišč.

2.7. Naj bo P množica n točk v ravnini. Označimo x- oziroma y-koordinato točke p s p_x in p_y. Točka $p \in P$ je *maksimalna*, če ne obstaja druga točka $q \in P$ s lastnostmi $p_x \leq q_x$ in $p_y \leq q_y$. Opiši algoritem, ki v času $O(n \log n)$ vrne podmnožico maksimalnih točk.

2.8. Naj bosta P in Q x-monotoni poti z n oziroma m oglišči. Predpostavimo, da niso nobena 3 oglišča kolinearna.

(a) Dokaži, da je moč $P \cap Q$ največ $O(n+m)$.

(b) Opiši algoritem, ki vrne $P \cap Q$ v linearnem času.

2.9. Naj bo S množica n paroma disjunktnih daljic v ravnini in naj bo p točka, ki ni na nobeni daljici iz S. Naj bo $S(p)$ podmnožica daljic iz S, ki so delno vidne iz p. To je, daljica $s \in S$ je v $S(p)$, če in samo če obstaja točka $q \in s$ z lastnostjo $\overline{pq} \cap s' = \emptyset$ za vsak $s' \in S \setminus \{s\}$. Opiši algoritem, ki vrne $S(p)$. (Nasvet: algoritem pometanja lahko pometa z nečim, kar ni premica ...)

2.10. Naj bo $M = \{m_1, m_2, \ldots, m_n\}$ množica n modrih trikotnikov v ravnini in naj bo $R = \{r_1, r_2, \ldots, r_n\}$ množica n rdečih trikotnikov v ravnini. Za parameter $t \in \mathbb{R}$ označimo z N_t navpično premico $\{(x,y) \in \mathbb{R}^2 \mid x = t\}$ in z D_t diagonalno premico $\{(x,y) \in \mathbb{R}^2 \mid x + y = t\}$. Zanima nas naslednji optimizacijski problem:

$$\phi(M,R) := \max_{t \in \mathbb{R}} \quad |\{m_i \in M \mid N_t \cap m_i \neq \emptyset\}| \cdot |\{r_i \in R \mid D_t \cap r_i \neq \emptyset\}|$$

Opiši algoritem, ki vrne $\phi(M,R)$ v času $O(n \log n)$.

2.11. Podana je množica P z n točkami v ravnini, dodatno pa še točki $q, q' \notin P$. Iščemo krog D_{min}, ki vsebuje q, q', pa čim manjše število točk množice P.

(a) Dokaži naslednjo trditev: če obstaja krog, ki vsebuje q, q' in še k točk iz P, potem obstaja tudi krog, ki ima q, q' na svojem robu in vsebuje največ k točk iz P.

(b) Za podano točko p naj bo $C(p)$ množica središč krogov, ki imajo q, q' na svojem robu in vsebujejo p. Kakšna je $C(p)$?

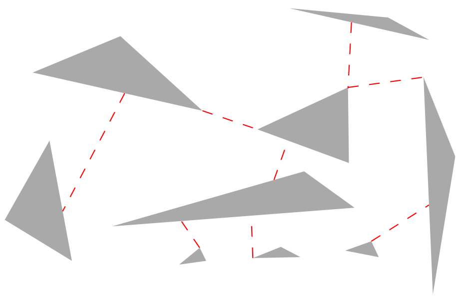
(c) Opiši algoritem, ki vrne krog D_{min} v času $O(n \log n)$.

2.12. Naj bo S množica n intervalov na realni osi $\mathbb{R}$. Predpostavimo naslednji splošni položaj: vse skrajne točke intervalov so različne. Opiši algoritem, ki v času $O(n \log n)$ prešteje elemente množice $\{(I, J) \in S^2 \mid I \subset J \text{ or } J \subset I\}$.

2.13. Podana je množica S z n paroma disjunktnimi trikotniki v ravnini. Želimo najti $n-1$ daljic z naslednjimi lastnostmi:

- Vsaka daljica povezuje točko na robu trikotnika s točko na robu nekega drugega trikotnika.
- Notrajnosti daljic so paroma disjunktne in disjunktne s trikotniki.
- Daljice povezujejo vse trikotnike, to je, obstaja pot od poljubnega trikotnika do vsakega drugega trikotnika, ki gre samo po daljicah in trikotnikih.

Ideja je, da želimo najti daljice, ki definirajo vpeto drevo vhodnih trikotnikov brez križanja. Glej sliko 2.6 Opiši algoritem, ki rešuje ta problem v času $O(n \log n)$.



Slika 2.6: Zgled za vajo 2.13.

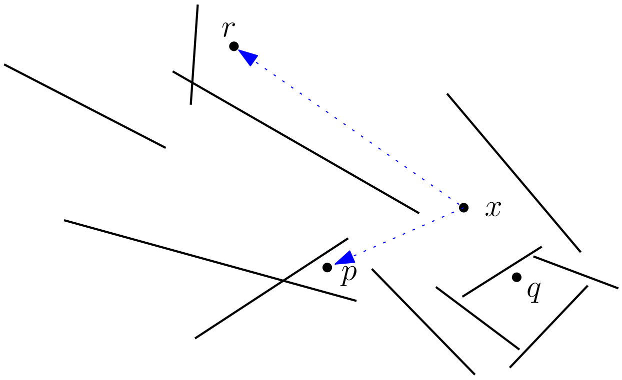
2.14. Naj bo S množica n premic in naj bo D krog v ravnini. Opiši algoritem, ki v času $O(n \log n)$ šteje, koliko parov premic ima presečišče v notranji D. (Nasvet: kdaj se dve premici sekata v notranji D?)

2.15. Naj bo S množica n daljic v ravnini, ki niso nujno disjunktne, in naj bosta p, q točki v ravnini. Opiši algoritem, ki določi, ali obstaja točka x v ravnini, ki vidi p in q. Glej sliko 2.7.

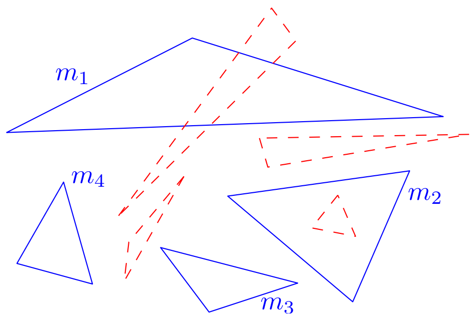
2.16. Naj bo M množica n modrih trikotnikov v ravnini, ki so paroma disjunktni. Naj bo R množica n rdečih trikotnikov v ravnini, ki so tudi paroma disjunktni. Naj bo X unija trikotnikov iz R. To je $X = \bigcup_{r \in R} r$. Opiši algoritem pometanja, ki v času $O(n \log n)$ izračuna

$$\{m \in M \mid m \cap X = \emptyset\}.$$

Torej iščemo trikotnike iz M, ki so disjunktni z R. Glej sliko 2.8.



Slika 2.7: Zgled za vajo 2.15, pri čemer ne obstaja točka, ki vidi p, q, ampak obstaja točka x, ki vidi p, r.



Slika 2.8: Zgled za vajo 2.16, pri čemer bi bil izhod $\{m_3, m_4\}$.

2.17. Naj bo $M = \{T_1, T_2, \ldots, T_n\}$ množica n trikotnikov v ravnini. Zanima nas naslednji optimizacijski problem.

$$\begin{aligned} \phi(M) := \max \quad & |\{T_i \in M \mid \ell_1 \cap T_i \neq \emptyset \text{ ali } \ell_2 \cap T_i \neq \emptyset\}| \\ \text{p.p.} \quad & \ell_1, \ell_2 \text{ sta navpični premici} \\ & \text{razdalja med } \ell_1 \text{ in } \ell_2 \text{ je enaka } 1 \end{aligned}$$

Opiši algoritem, ki vrne $\phi(M)$ v času $O(n \log n)$.

2.18. Naj bo P množica n točk v ravnini. Zanima nas naslednji optimizacijski problem.

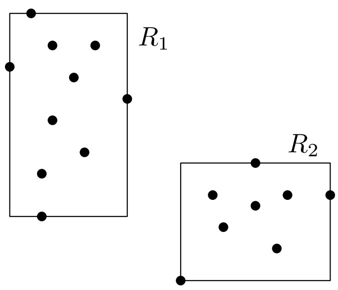
$$\begin{aligned} \phi(P) := \min \quad & \texttt{ploščina}(R_1) + \texttt{ploščina}(R_2) \\ \text{p.p.} \quad & R_1, R_2 \text{ sta pravokotnika} \\ & p_x < q_x \text{ za vse } (p_x, p_y) \in R_1,\ (q_x, q_y) \in R_2 \\ & P \subset R_1 \cup R_2 \end{aligned}$$

Glej sliko 2.9 za zgled. Opiši algoritem pometanja, ki vrne $\phi(P)$ v času $O(n \log n)$.

2.19. Naj bo R množica n rdečih kvadratov v ravnini in naj bo M množica n modrih kvadratov v ravnini. Vsi kvadrati iz R in M imajo isto velikost in imajo robove vzporedne s koordinatnima osema. Kvadrati iz R so paroma disjunktni in kvadrati iz M so paroma disjunktni. Zanima nas množica

$$Pre(R, M) = \{(r, m) \in R \times M \mid r \cap m \neq \emptyset\}.$$

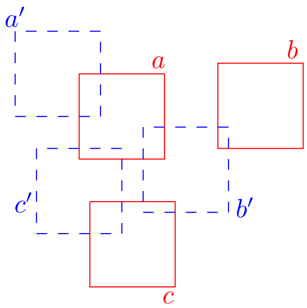
Glej sliko 2.10.



Slika 2.9: Točke P in pravokotnika R_1,R_2, ki definirata dopustno rešitev pri vaji 2.18.

(a) Dokaži, da je moč množice $Pre(R, M)$ največ $O(n)$.

(b) Opiši algoritem pometanja, ki vrne $Pre(R, M)$ v času $O(n \log n)$.



$R = \{a, b, c\}$
$M = \{a', b', c'\}$
$Pre(R, M) = \{(a, a'), (a, b'), (a, c'), (b, b'), (c, b'), (c, c')\}$

Slika 2.10: Zgled za vajo 2.19.

2.20. *Vertikalna razdalja* $d_v(s, s')$ med daljicama s, s' je

$$d_v(s, s') = \min\{|p_y - p'_y| \mid (a, p_y) \in s, (a, p'_y) \in s', a \in \mathbb{R}\}.$$

Povedano drugače, vertikalna razdalja med s, s' je dolžina najkrajše navpične daljice, ki seka s in s'. Če s in s' nimata točk z isto x-koordinato, potem je $d_v(s, s')$ nedefinirana (ali $+\infty$).

Naj bo S množica n daljic v ravnini. Daljice iz S so paroma disjunktne. Opiši output-sensitive algoritem, ki vrne množico

$$\{(s, s') \in S^2 \mid d_v(s, s') \leq 1\}.$$

2.21. Podana je množica $P = \{p_1, p_2, p_3, \ldots, p_n\}$ n točk v ravnini. Označimo z R najmanjši pravokotnik, ki ima stranice vzporedne s koordinatnimi osmi in vsebuje P. Opiši algoritem pometanja, ki v času $O(n \log n)$ najde kvadrat velikosti 1, ki je vsebovan v R, vendar sam ne vsebuje nobene točke iz P; če pa tak kvadrat ne obstaja, naj algoritem to javi. Stranice kvadrata morajo biti vzporedne koordinatnim osem.

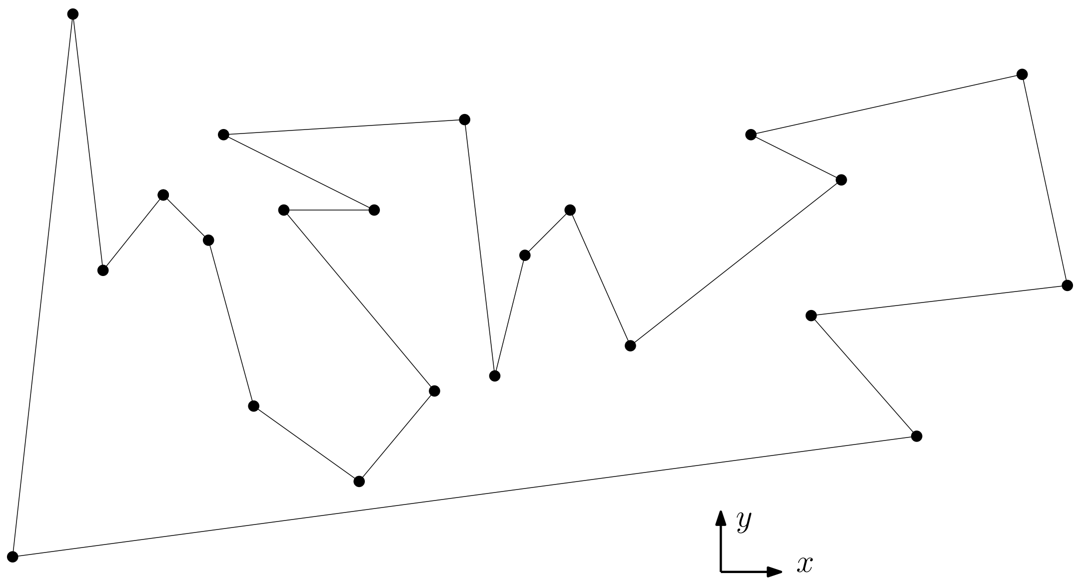
2.22. Imamo 42 enako velikih pravokotnih rezin sira; vsaka rezina ima n kvadratnih lukenj velikosti 1, vzporednih stranicam rezine. Rezine zložimo na sendvič eno na drugo tako, da so njihovi robovi skladni in da natanko prekrijejo kruh. Položaji lukenj glede na globalen koordinatni sistem so pri tem znani. Na vrh sendviča namažemo majonezo. Ali obstaja kakšna točka, v kateri imajo vse rezine luknjo in lahko zato majoneza pricurlja do kruha? Opiši postopek, ki najde odgovor v času $O(n \log n)$.

2.23. Podana je smer v v ravnini ter m disjunktnih konveksnih večkotnikov s skupno n oglišči. Opiši algoritem, ki v času $O(n \log n)$ določi vse večkotnike, ki so vsaj delno vidni z neskončne razdalje v smeri v. Večkotnik P je delno viden, če obstaja poltrak v smeri v s krajiščem na robu P, ki ne seka nobenega drugega večkotnika.

3 Večkotniki in subdivizije

Ko nič ne rečemo, so večkotniki podani kot seznam oglišč, naštetih v smeri urinega kazalca. Predpostaviš lahko, da je seznama implementiran s poljem (array) ali pa s kazalci – kar ti pri reševanju bolj ustreza.

3.1. V slikah 3.11-3.12 nariši diagonale, ki jih izračuna algoritem za triangulacijo, ki ga si spoznal na predavanjih.



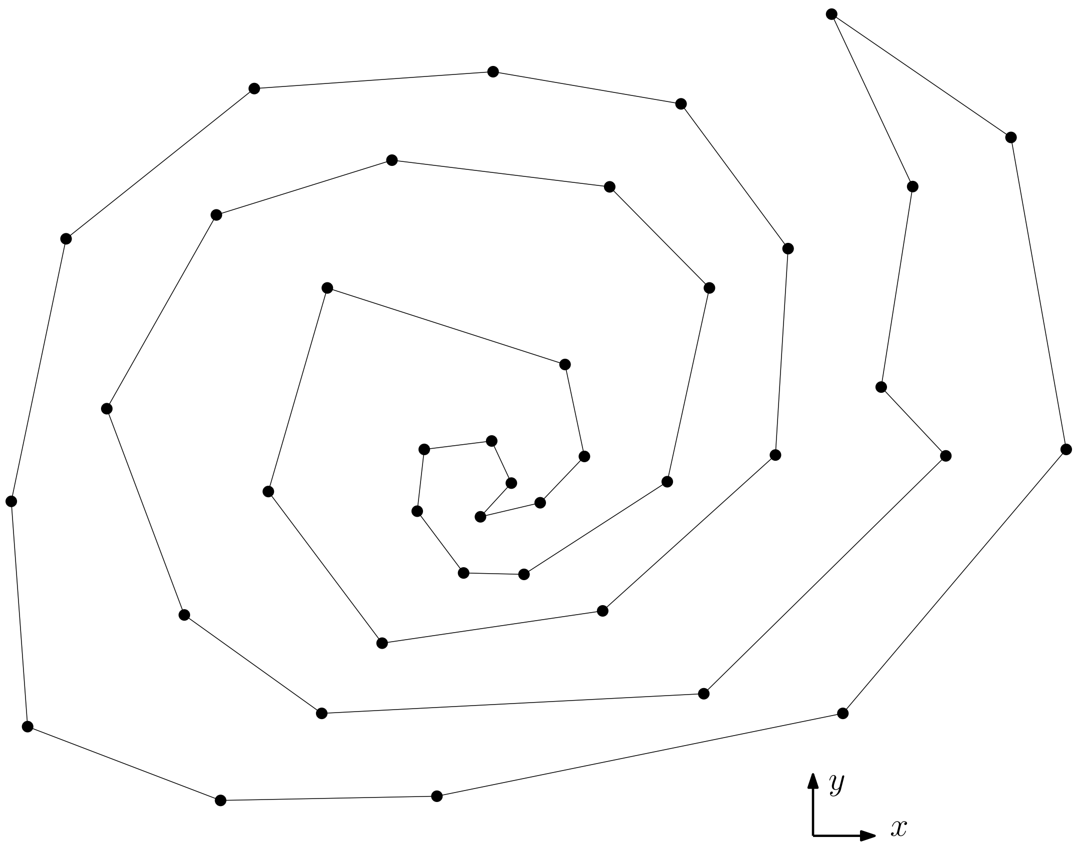
Slika 3.11: Vaja 3.1.

3.2. Podan je mnogokotnik z naslednjimi oglišči:

$$p[0] = (a, 4), \quad p[1] = (6, 0), \quad p[2] = (3, 0), \quad p[3] = (2, 9),$$
$$p[4] = (4, 6), \quad p[5] = (7, 5), \quad p[6] = (10, 5), \quad p[7] = (13, 6),$$
$$p[8] = (14, 10), \quad p[9] = (15, 0), \quad p[10] = (12, 4), \quad p[11] = (8, 0),$$

pri čemer velja $a \in [5, 11]$. Pomagaj si s sliko 3.13.

Triangulacijo mnogokotnika izračunamo z uporabo algoritma, ki si ga spoznal na predavanjih. Poišči 3 vrednosti parametra a (točke mnogokotnika morajo biti v splošnem položaju) pri katerih vrne algoritem različne triangulacije. Opazi, da lahko algoritem isto triangulacijo izračuna na različne načine. Za te 3 izbrane vrednosti parametra a nariši diagonale, ki jih algoritem izračuna. Nariši tri ločene slike!



Slika 3.12: Vaja 3.1.

3.3. Večkotnik na sliki 3.14 želimo triangulirati z algoritmom s predavanj. Nariši maksimalno območje, za katerega velja: če premikamo točko p znotraj tega območja, bo rezultat algoritma za triangulacijo ostal nespremenjen, vključno z vrstnim redom, v katerem algoritem vstavlja diagonale.

3.4. Podan je večkotnik kot seznam oglišč, vendar ni znano, ali so oglišča našteta v smeri urnega kazalca ali nasprotni smeri. Opiši algoritem, ki v linearnem času ugotovi smer.

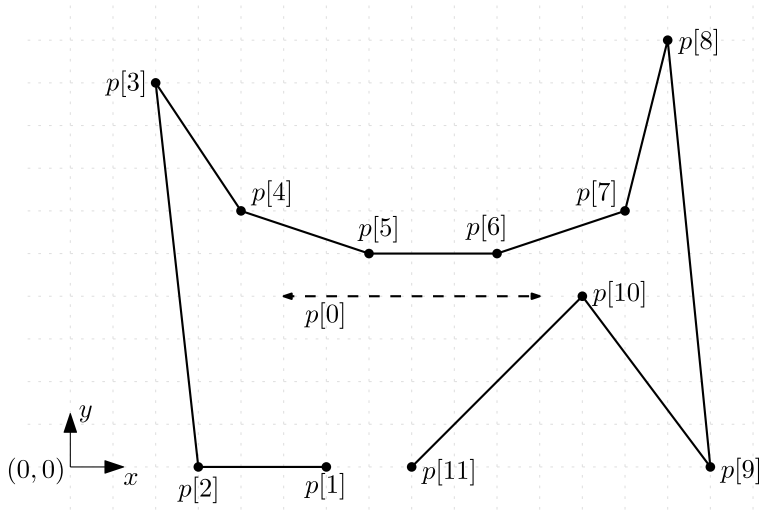
3.5. Kaj je narobe v naslednjem dokazu obstoja diagonale v večkotniku?

> Z uporabo zasuka lahko predpostavimo, da imajo oglišča večkotnika različne x-koordinate. Naj bo p oglišče večkotnika, ki ima najmanjšo x-koordinato. Torej je p konveksno oglišče večkotnika. Naj bo p_- oglisče pred p v seznamu oglišč večkotnika, in naj bo p_+ oglišče po p. Če je daljica $\overline{p_-p_+}$ diagonala, smo končali. Sicer naj bo q oglišče večkotnika, ki je v trikotniku $\triangle(p, p_-, p_+)$ in minimizira razdaljo od p, pa ni p. Potem je daljica $\overline{pq}$ diagonala.

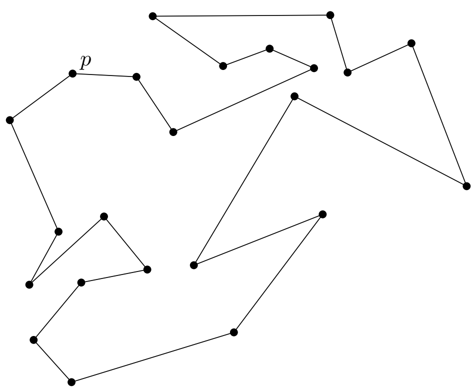
3.6. Zapiši psevdokodo algoritma, ki izračuna 3-barvanje vhodne triangulacije večkotnika v linearnem času.

3.7. Opiši algoritem, ki vrne diagonalo vhodnega večkotnika v linearnem času.

3.8. Dokaži ali ovrži:



Slika 3.13: Vaja 3.2.



Slika 3.14: Vaja 3.3.

(a) Naj bo P x-monoton večkotnik in naj bo T triangulacija večkotnika P. Potem je dualen graf od T pot.

(b) Naj bo P x-monoton večkotnik. Potem za P obstaja triangulacijo, katere dualen graf je pot.

(c) Za vsako drevo D obstaja večkotnik s triangulacijo, ki ima D kot dualen graf.

(d) Za vsako drevo D, ki ima največjo stopnjo 3, obstaja konveksen večkotnik s triangulacijo, ki ima D kot dualen graf.

(e) Kakšno lastnost drevesa lahko dodamo v (d), da je trditev pravilna.

(f) Obstajajo večkotniki, ki imajo samo eno triangulacijo.

(g) Obstajajo večkotniki, ki so monotoni za samo eno smer.

(h) Vsako konveksno oglišče x-monotonega večkotnika, različno od x-skrajnih točk, je vogal ušesa.

(i) Ali je (h) pravilen za *mountain polygons*, to je, za x-monoton večkotnik, pri katerem je spodnja poligonalna pot daljica.

(j) Vsak večkotnik, ki ima sodo število robov, lahko z diagonalami razdelimo na četverokotnike.

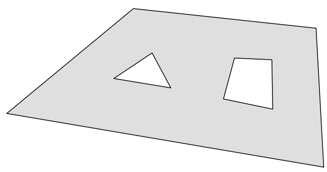
(k) Zbodno število triangulacije večkotnika je največje število diagonal, ki jih lahko preseka poljubna premica v ravnini. Ali obtajajo večkotniki, pri katerih ima vsaka triangulacija zbodno število reda $\Omega(n)$?

3.9. Naj bo P večkotnik in naj bo S množica točk. Opiši algoritem pometanja, ki v času $O(n \log n)$ vrne $S \cap P$, kjer je n velikost vhoda (število oglišč P plus $|S|$).

3.10. Večkotnik P je *zvezden*, če obstaja točka $p \in P$, ki vidi cel večkotnik. Podana sta zvezden večkotnik P in točka p, ki vidi cel P. Opiši algoritem, ki vrne v linearnem času triangulacijo večkotnika P.

3.11. Podani so večkotnik P in točki $s, t \in P$. Opiši algoritem, ki vrne najkrajšo pot, vsebovano v P, od s do t. (Namig: razmisli prej, kako izgleda najkrajša pot.)

3.12. Naj bo D poligonalna domena s n oglišč in h luknjami. Koliko trikotnikov ima triangulacija domene D? Podrobno dokaži pravilnost odgovora. Glej sliko 3.15.



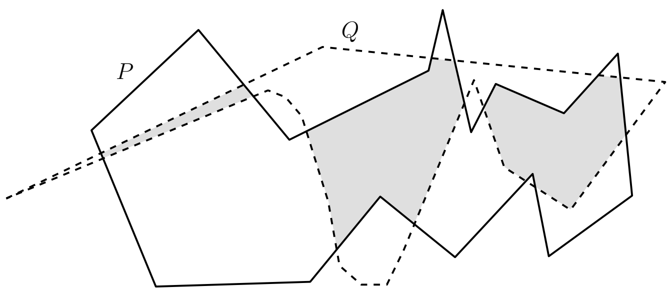
Slika 3.15: Zgled za vajo 3.12, pri čemer ima domena $n = 11$ oglišč in $h = 2$ lukenj.

3.13. Naj bosta P in Q x-monotona mnogokotnika s skupno n oglišči. Glej sliko 3.16.

(a) Dokaži, da ima $P \cap Q$ največ $O(n)$ povezanih komponent.

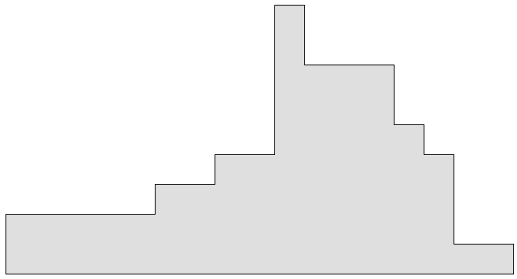
(b) Opiši algoritem, ki prešteje, koliko povezanih komponent ima $P \cap Q$.

(b) Če P je x-monoton in Q y-monoton (in ne nujno tudi x-monoton), koliko povezanih komponent ima lahko $P \cap Q$? Zakaj?



Slika 3.16: Zgled za vajo 3.13, pri čemer ima presek 3 povezane komponente.

3.14. Podana sta večkotnik P z n ali manj oglišč in množica S daljic moči manjši ali enako n. Vemo, da so daljice v S paroma disjunktne. Opiši algoritem pometanja, ki v času $O(n \log n)$ vrne daljice iz S, ki so vsebovane v P.



Slika 3.17: Pravokotna piramida.

3.15. Opiši učinkovit algoritem za naslednji problem: podana sta večkotnik P in notranja točka $p \in P$, nas pa zanima konstrukcija območja znotraj večkotnika P, ki je vidno iz točke p.

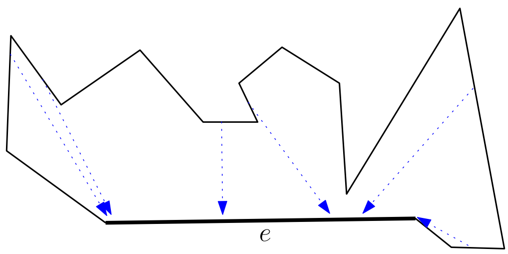
3.16. Večkotnik je pravokoten, če je vsak rob vodoraven ali navpičen. Pravokotna piramida je pravokoten večkotnik, ki je x-monoton in pri katerem je spodnja pot daljica, zgornja pot pa sestavljena iz dveh y-monotonih poti. Glej sliko 3.17. Predpostavimo, da noben par robov ni kolinearen.

(a) Dokaži naslednjo trditev: vsako pravokotno piramido lahko razdelimo z diagonalami na konveksne četverokotnike.

(b) Opiši algoritem, ki v linearnem času najde diagonale iz točke (a).

3.17. Podana sta n-kotnik P ter točka p_0, ki sovpada z enim od P-jevih oglišč. Opiši algoritem, ki v času $O(n \log n)$ najde takšno triangulacijo večkotnika P, ki maksimizira število diagonal z enim od krajišč v p_0.

3.18. Večkotnik P je *viden z roba* e, če vidi vsaka točka $q \in P$ kakšno točko na e. Glej sliko 3.18. Podan je večkotnik P in rob e, s katerega je P viden. Opiši algoritem, ki v linearnem času vrne triangulacijo večkotnika P. Razloži vlogo vidnosti z roba.



Slika 3.18: Zgled za vajo 3.18, pri čemer je večkotnik viden iz roba e.

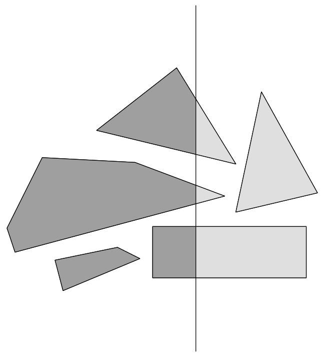
3.19. Cilj te vaje je algoritem, ki najde diagonalo, ki razdeli večkotnik na dva kosa, ki sta velika.

(a) Naj bo T drevo z m vozliščami in največjo stopnjo 3. Pokaži, da obstaja povezava e tako, da ima vsako drevo v $T - e$ največ $m/3$ vozlišč.

(b) Naj bo P podan večkotnik z n ogliščami. Pokaži, da obstaja diagonala, ki deli P na dva večkotnika, ki imata največ $2n/3 + 2$ oglišč.

(c) Opiši algoritem, ki najde diagonalo točke (b) v času $O(n \log n)$.

3.20. Podani so konveksni disjunktni večkotniki $P_1, P_2, \ldots, P_k$ v ravnini. Vsi večkotniki skupaj imajo n oglišč. Opiši algoritem, ki v času $O(n \log n)$ najde navpično premico ℓ z naslednjo lastnostjo: ploščina dela $P_1 \cup \cdots \cup P_k$ levo od premice ℓ in ploščina dela $P_1 \cup \cdots \cup P_k$ desno od premice ℓ sta enaki. Glej sliko 3.19.
Ali je v tvojem algoritmu pomembno, da so podani večkotniki konveksni in disjunktni?



Slika 3.19: Zgled za vajo 3.20.

3.21. Naj bo P večkotnik v ravnini z n oglišč. Opiši algoritem, ki v času $O(n \log n)$ najde navpično premico ℓ z naslednjo lastnostjo: ploščina dela večkotnika levo od premice ℓ in ploščina dela večkotnika desno od premice ℓ sta enaki.

3.22. (Težje) Opiši algoritem, ki v linearnem času določi, ali obstaja smer, v kateri je vhodni večkotnik monoton.

4 Konveksnost

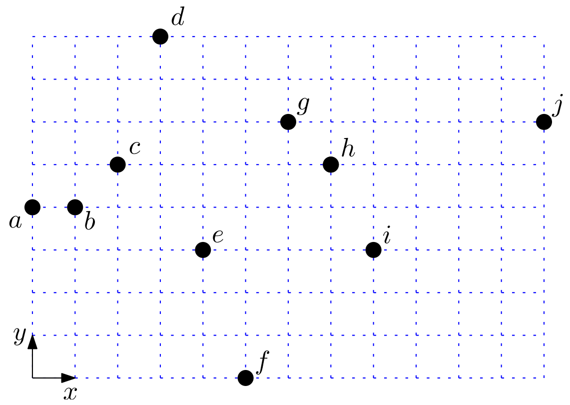
4.1. Množica desetih točk $\mathcal{P} = \{a, b, \ldots, j\}$ je podana na slikah 4.20 in 4.21. Uporabljamo prirastni algoritem za računanje zgornje lupine točk $\mathcal{P}$. Navedi trojke točk $(X, Y, Z) \in \mathcal{P}^3$, za katere algoritem preveri, ali je zaporedje X, Y, Z zasuk na desno. V seznamu morajo biti trojke v istem vrstnem redu, kot jih algoritem preveri.

Povej tudi seznam trojk pri Grahamovem algoritmu.

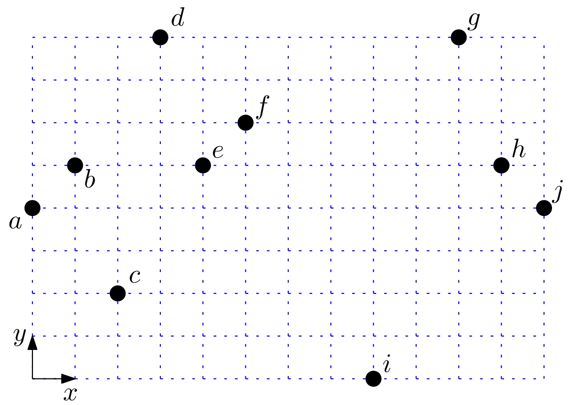
4.2. Naj bosta $A, B \subset \mathbb{R}^d$ konveksni množici. Ugotovi, ali X konveksna množica, ko je

- $X = A \setminus B$,
- $X = A \cup B$,
- $X = A + B = \{a + b \mid a \in A, b \in B\}$,
- $X = A \times B$.

4.3. Določi konveksno lupino množice $\{(0,0)\} \cup \{(1,t) \in \mathbb{R}^2 \mid t \in (0,1)\}$ in nato še množice $\{(x,y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1 \text{ ali } (x-3)^2 + y^2 < 1\}$.



Slika 4.20: Za vajo 4.1.



Slika 4.21: Za vajo 4.1.

4.4. Za splošen n opiši primer množice n točk, v katerem vsaj eno njegovo oglišče nastopa v $O(n)$ trojicah, ki jih preveri prirastni algoritem za izračun konveksne lupine.

4.5. Videli smo Grahamov algoritem, ki je sestavljen iz dveh korakov: (1) iz podanih točk na poseben način zgradimo večkotnik M in se potem (2) sprehajamo po M in ga spreminjamo, dokler ne dobimo $CH(P)$. Dokaži, da samo korak (2) ni dovolj za izračun konveksne lupine večkotnika: najdi tak večkotnik N, da ne dobimo $CH(N)$, če na njem direktno uporabimo korak (2) Grahamovega algoritma.

4.6. Naj bo P množica točk v ravnini. Za vsako točko $p \in P$ definiramo

$$cell(p) = \{x \in \mathbb{R}^2 \mid d(x,p) \leq d(x,p') \text{ za vsak } p' \in P\},$$

kjer je $d(\cdot,\cdot)$ evklidska razdalja. Pokaži, da je $cell(p)$ konveksna množica za vsako točko $p \in P$.

4.7. Opiši podatkovno strukturo za hranjenje konveksnega večkotnika P, ki omogoča v času $O(\log n)$ ugotoviti, ali je poizvedbena točka vsebovana v P. Konstrukcija podatkovne strukture mora biti izvedena v linearnem času.

4.8. Naj bo P končna množica točk v ravnini. Dokaži naslednjo trditev: $CH(P)$ je konveksna množica, ki vsebuje P in ima najmanjšo ploščino.

4.9. Podana je končna množica točk P v ravnini. Naj bo M najkrajši večkotnik, ki vsebuje P.

(a) Dokaži, da je M konveksen večkotnik.

(b) Dokaži, da je $M \subseteq Q$ za poljubno konveksno množico Q, ki vsebuje P.

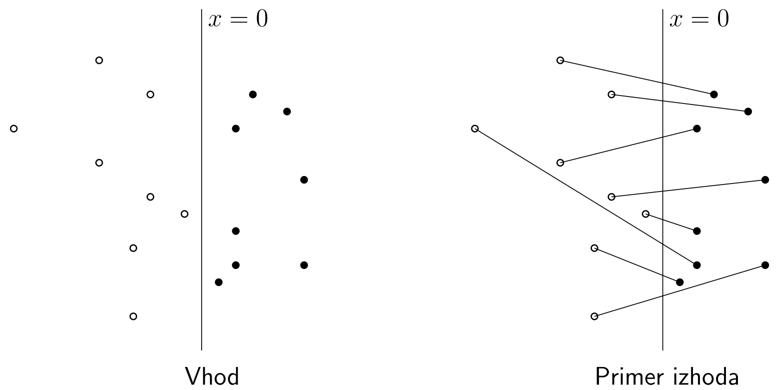
(c) Dokaži, da je $CH(P) = M$.

4.10. Konveksen večkotnik M je podan kot seznam oglišč, naštetih v smeri urinega kazalca. Opiši algoritem, ki uredi oglišča od M po x-koordinatah v linearnem času. Ali se da tvoj algoritem ustrezno dopolniti, da bo (še vedno v linearnem času) deloval za poljuben večkotnik M?

4.11. Naj bo P množica oglišč večkotnika M. Ali je res, da $CH(P) = CH(M)$?

4.12. Podana sta dva konveksna večkotnika P, Q. Opiši algoritem, ki izračuna $CH(P \cup Q)$ v linearnem času. (Če ne gre drugače, lahko predpostaviš, da sta P in Q disjunktna.)

4.13. Podani sta množica točk L na levi od premice $x = 0$ in množica točk D na desni od $x = 0$. Množici L in D imata isto moč n. Predpostavimo, da niso nobene 3 točke iz $D \cup L$ na isti premici. Prirejanje $M \subset L \times D$ med L in D je *disjunktno prirejanje*, če so daljice $\overline{ld}$, $(l, d) \in M$ paroma disjunktne. Glej sliko 4.22. Opiši algoritem, ki vrne disjunktno prirejanje M v času $O(n^2 \log n)$.



Slika 4.22: Zgled za vajo 4.13.

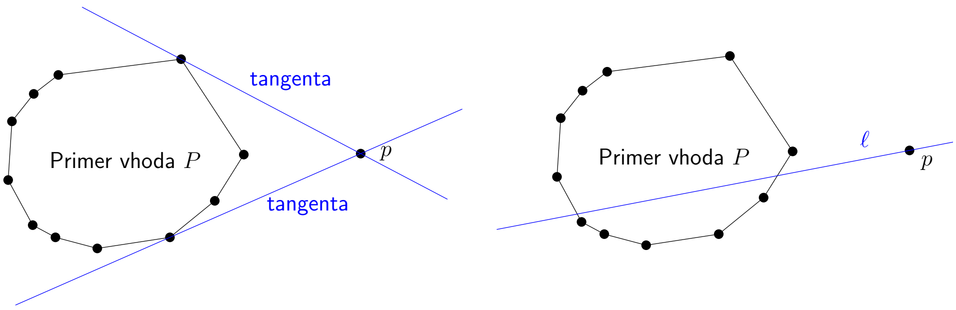
4.14. Opazujemo problem v prvem delu vaje 1.2. Pokaži, da je spodnja meja računske zahtevnosti problema $\Omega(n \log n)$.

4.15. Podan je konveksen večkotnik P kot polje (array) oglišč $p[1, \ldots, n]$.

(a) Opiši algoritem, ki v času $O(\log n)$ vrne najvišje oglišče iz P.

(b) Opiši algoritem, ki v času $O(\log n)$ vrne obe tangenti od podane točke $p \in \mathbb{R}^2 \setminus P$. Glej levo stran slike 4.23.

(c) Opiši algoritem, ki v času $O(\log n)$ vrne presečišče med P in podano premico ℓ. Glej desno stran slike 4.23.



Slika 4.23: Zgled za vajo 4.15.

(d) Opiši algoritem, ki v času $O(\log n)$ vrne točko od P, ki je najbljžje podani premici ℓ.

4.16. Podani sta množica točk P in točka q v ravnini. Opiši algoritem, ki v linearnem času odloči, ali je q v notranji $CH(P)$.

4.17. Premer $\Delta(P)$ množice točk P je $\Delta(P) = \max\{d(p,q) \mid p,q \in P\}$, kjer je $d(\cdot,\cdot)$ evklidska razdalja. Naj bo P končna množica n točk.

- Dokaži, da je $\Delta(P) = \Delta(Q)$, kjer je Q množica oglišč iz $CH(P)$.
- Dokaži, da se pojavi vrednost $\Delta(P)$ največ $O(n)$-krat v seznamu $d(p,q), p,q \in P$.
- Opiši algoritem, ki izračuna $\Delta(P)$ v času $O(n \log n)$.

4.18. Naj bodo $P_1, P_2, \ldots, P_t$ množice točk v $\mathbb{R}^d$, in $K_i = CH(P_i)$ $(i = 1, 2, \ldots, t)$ njihove konveksne ovojnice. Definirajmo še

$$P = \bigcup_{i=1}^{t} P_i \quad \text{in} \quad K = \bigcup_{i=1}^{t} K_i.$$

Ugotovi, ali je $CH(P) = CH(K)$.

4.19. Naj bo P večkotnik, in naj bo

$$K(P) = \{p \in P \mid p \text{ vidi cel večkotnik } P\}.$$

Pokaži, da je $K(P)$ konveksna množica.

4.20. Naj bo P množica točk v $\mathbb{R}^3$. Definiramo

$$U(P) = \{x \in \mathbb{R}^3 \mid d(x,p) \leq 1 \text{ za vsak } p \in P\},$$

kjer je $d(\cdot,\cdot)$ evklidska razdalja. Pokaži, da je $U(P)$ konveksna množica.

4.21. Naj bo $P = \{p_1, \ldots, p_n\}$ množica točk v $\mathbb{R}^2$. $CH(P)$ smo definirali kot množico točk $x \in \mathbb{R}^2$, ki zadovolji

$$x = \sum_{i=1}^{n} \lambda_i p_i, \; \sum_{i=1}^{n} \lambda_i = 1, \; \lambda_1, \ldots, \lambda_n \geq 0.$$

Kaj se zgodi, če odstranimo pogoj $\lambda_1, \ldots, \lambda_n \geq 0$? Vedno?

4.22. Naj bo P množica točk v $\mathbb{R}^2$. $CH^*(P)$ definiramo kot množico

$$\{x \in \mathbb{R}^2 \mid x = \lambda p + (1-\lambda)q,\ \lambda \in [0,1],\ p, q \in P\}.$$

Navedi primer, ki pokaže $CH(P) \neq CH^*(P)$. Ali je $CH(P) = CH^*(CH^*(P))$?

4.23. Naj bo P končna množica točk v ravnini in naj bo M konveksen večkotnik, ki vsebuje P in ima oglišča iz P. Dokaži, da je $M = CH(P)$.

4.24. Naj bosta P, Q konveksna večkotnika. Opiši algoritem, ki vrne $P \cap Q$ v linearnem času.

4.25. Podana sta dva konveksna večkotnika M, M' kot seznama oglišč. Opiši algoritem, ki v linearnem času odloči, ali velja $M \subset M'$.

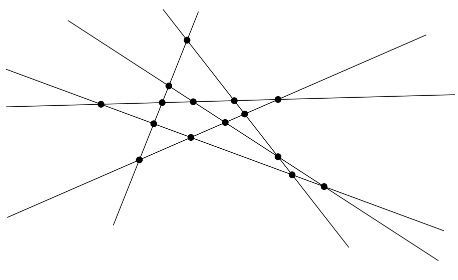
4.26. Opazujemo problem v vaji 1.6. Pokaži, da je spodnja meja računske zahtevnosti problema $\Omega(n \log n)$.

4.27. Opazujemo problem v vaji 1.10. Pokaži, da je spodnja meja računske zahtevnosti problema $\Omega(n \log n)$.

4.28. Podana sta krog D ter konveksen n-kotnik P, ki je v celoti vsebovan v D. Iščemo točko $x \in D \backslash P$ (torej znotraj kroga, vendar zunaj večkotnika), iz katere je vidno največje možno število P-jevih oglišč. S psevdokodo opiši algoritem, ki najde točko x v času $O(n)$.

4.29. Podan je konveksen n-kotnik P. Iščemo diagonalo d, ki razdeli P na dva kosa, katerih razlika ploščin je po absolutni vrednosti najmanjša možna. Zapiši psevdokodo za algoritem, ki najde takšno diagonalo v linearnem času.

4.30. Naj bo $L = \{\ell_1, \ell_2, \ldots, \ell_n)$ množica n premic v ravnini v splošnem položaju (nobeni 2 premici nista vzporedni in presek poljubnih 3 premic je prazen). Naj bo I množica presečišč premic iz L, to je $I = \{\ell_i \cap \ell_j \mid \ell_i \in L, \ell_j \in L,\ i \neq j\}$. Glej sliko 4.24. Opiši (s psevdokodo) algoritem, ki vrne konveksno lupino $CH(I)$ v času $O(n \log n)$.



Slika 4.24: Množica premic L in množica presečišč I.

4.31. Podan je konveksen večkotnik P. Naj bosta $E(P)$ in $V(P)$ množica robov oziroma oglišč večkotnika P. Naj bo $\mathcal{T}$ množica trikotnikov, ki imajo oglišča v $V(P)$ in najmanj en rob v $E(P)$. Opiši algoritem, ki v linearnem času vrne trikotnik v $\mathcal{T}$ z največjo ploščino.

4.32. Podana je množica P z n točkami v ravnini. Zanima nas algoritem, ki najde pravokotnik, ki vsebuje P in ima najmanjšo ploščino. Glej sliko 4.25.

(a) Pokaži, da pravokotnik vsebuje P natanko tedaj, ko vsebuje $CH(P)$.

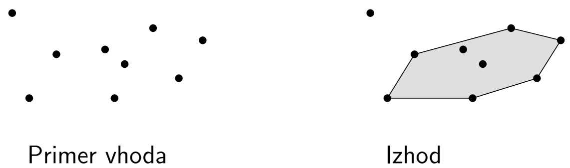
(b) Pokaži, da je dovolj, če iščemo samo med pravokotniki, ki imajo en rob istoležen z enim od robov konveksne lupine $CH(P)$. Nasvet: trigonometrija.

(c) Pokaži, da v času $O(n \log n)$ lahko razdelimo $\mathbb{S}^1$ na intervale $I_1, I_2, \ldots$ z naslednjo lastnostjo: skrajni točki v smeri $\vec{u}$ in $-\vec{u}$ sta isti za vsak $\vec{u} \in I_j$.

(d) Opiši algoritem, ki v času $O(n \log n)$ vrne pravokotnik, ki vsebuje P in ima najmanjšo ploščino

Slika 4.25: Pravokotnik, ki vsebuje množico točk P.

4.33. Podana je množica n točk P v ravnini. Naj bo M_{-1} večkotnik, ki vsebuje vse točke iz P razen ene točke (lahko izberemo, katera točka ni vključena) in ima najmanjši obseg. Glej sliko 4.26.



Slika 4.26: Zgled za vajo 4.33.

Opiši algoritem, ki najde M_{-1} v času $O(nh \log n)$, kjer je h število skrajnih točk v P. (V času $O(n^2 \log n)$ je lažje.)

(Težje) Predpostavimo, da imamo naslednji podprogram: vhod je n disjunktnih trikotnikov in n točk in podprogram vrne v času $O(n \log n)$, katere točke so v vsakem trikotniku. (Tak podprogram lahko implementiramo z uporabo algoritma pometanja.) Dokaži, da lahko najdemo M_{-1} v času $O(n \log n)$.

4.34. Naj bo $T(n)$ število različnih triangulacij konveksnega večkotnika z n oglišči. Opiši rekurzivno enačbo za $T(n)$.

(Težje) Kakšna je rešitev rekurzivne enačbe?

4.35. *Zbodno število* triangulacije večkotnika je največje število diagonal, ki jih lahko preseka poljubna premica v ravnini. Opiši algoritem, ki za vhodni konveksni večkotnik z n oglišči konstruira triangulacijo z zbodnim številom $O(\log n)$.

4.36. Zanimajo nas konveksne lupine množic n krogov.

(a) Predpostavi, da imajo vsi krogi isto velikost. Dokaži, da se lahko pojavi vsak krog največ enkrat na robu konveksne lupine.

(b) Dokaži, da se lahko posamezen krog v konveksni ovojnici pojavi $n-1$ krat, če imamo kroge različnih velikosti.

(c) (Težje) Dokaži, da ima konveksna lupina krogov različnih velikosti zahtevnost $O(n)$, to je, ima $O(n)$ daljic in krožnih lokov.

(d) (Težje) Opiši algoritem, ki v času $O(n \log n)$ skonstruira konveksno ovojnico n krogov različnih velikosti.

5 Podatkovne strukture za točke

V tem razdelku imajo vsi pravokotniki in kvadrati robove vzporedne s koordinatnimi osmi.

5.1. Podana je množica n točk P v ravnini. Poiskati želimo dinamično podatkovno strukturo, ki hrani podmnožico $Q \subseteq P$. Na začetku je $Q = P$. Podatkovna struktura mora podpirati naslednje operacije v času $O(\log^2 n)$:

- Brisanje točke $p \in Q$ iz Q.
- Vstavljanje točke $p \in P \setminus Q$ v Q.
- Štetje $|Q \cap R|$ za vhodni pravokotnik R (poizvedba).

5.2. Podana je množica n točk P v $\mathbb{R}^d$, kjer je d konstanta. Poiskati želimo podatkovno strukturo, ki hrani P in išče *delne zadetke*. Poizvedba za delne zadetke ima kot vhod vrednosti nekaterih koordinat, podatkovna struktura pa mora vrniti točke iz P, ki se v podanih koordinatah ujemajo s poizvedbo.

(a) Razloži, kako lahko delne zadetke v $\mathbb{R}^2$ najde 2-dimenzionalno območno drevo. Kakšen čas je potreben za odgovor?

(b) Opiši podatkovno strukturo, ki uporablja linearno mnogo prostora in najde delne zadetke v času $O(\log n + k)$. Ne pozabi, da lahko porabimo $O(d2^d n)$ prostora, ker je d konstanta.

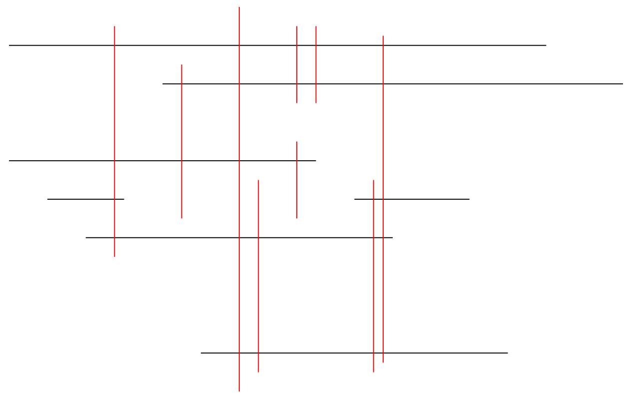
5.3. Naj bo H množica največ n vodoravnih daljic in naj bo V množica največ n navpičnih daljic. Cilj vaje je algoritem, ki v času $O(n \log n)$ določi število parov iz $H \times V$, ki se sekajo. Glej sliko 5.27.

(a) Naj bo P množica n točk v $\mathbb{R}$. Pokaži, da obstaja dinamična podatkovna struktura, ki shrani podmnožico $Q \subseteq P$ in v času $O(\log n)$ lahko dela naslednje operacije: vstavljanje elementa iz $P \setminus Q$, brisanje elementa iz Q, štetje točk $Q \cap I$ za podan interval I.

(b) Opiši algoritem, ki v času $O(n \log n)$ določi število parov iz $H \times V$, ki se sekajo.

5.4. Naj bo P množica n točk v ravnini. Opiši podatkovno strukturo za hranjenje P z naslednjimi lastnostmi:

- čas gradnje podatkovne strukture je $O(n \log n)$;



Slika 5.27: Zgled za vajo 5.3, pri čemer mora vrniti algoritem 26.

- za pravokotnik R (poizvedba) vrne podatkovna struktura v času $O(\log^2 n)$ najmanjši pravokotnik, ki vsebuje vse točke iz $P \cap R$.

5.5. Naj bo $\mathcal{R}$ množica trikotnikov v ravnini, ki so določeni z neenačbami tipa $x \geq a, y \geq b, x + y \leq c$, $a, b, c \in \mathbb{R}$. Opiši podatkovno strukturo, ki hrani P in odgovarja na naslednje poizvedbe: za podan $T \in \mathcal{R}$ vrne $T \cap P$. Čas gradnje podatkovne strukture mora biti $O(n \log^2 n)$ in čas odgovarjanja na posamezno poizvedbo mora biti $O(|T \cap P| + \log^3 n)$. Nasvet: uporabljaj več kot 2 dimenziji.

5.6. Dokaži, da je analiza časa odgovora v kd-drevesu tesna. To je, opiši množico točk P in pravokotnik R tako, da potrebuje kd-drevo $\Omega(\sqrt{n})$ časa za odločanje, ali je $P \cap Q = \emptyset$.

5.7. Koliko časa potrebuje kd-drevo, če ga vprašamo za točke v območju $[a, a] \times [b, b]$? In koliko potrebuje območno drevo?

5.8. Naj bo Q množica enako velikih kvadratov v $\mathbb{R}^2$. Opiši učinkovito podatkovno strukturo, ki hrani Q in hitro odgovori na poizvedbe naslednje oblike: kateri kvadrati iz Q vsebujejo podano vhodno točko.

5.9. Podana je množica n točk P na premici $\mathbb{R}$. Opiši podatkovno strukturo z lastnostmi:

- čas gradnje je $O(n \log n)$;
- podatkovna struktura porabi $O(n)$ prostora;
- za podani interval I vrne podatkovna struktura naključno točko iz $P \cap I$ v času $O(\log n)$. To je, vsaka točka iz $P \cap I$ ima isto verjetnost, da je v odgovoru.

Opiši podobno strukturo za $\mathbb{R}^2$ z malo večjo porabo prostora in časa.

5.10. Podana je množica n točk P v $\mathbb{R}$. Vsaka točka $p \in P$ ima težo $w_p \in \mathbb{R}$. Želimo podatkovno strukturo za hraniti P, ki ga lahko konstruiramo v času $O(n \log n)$ in ki vrne za vhodni interval R (poizvedba) točke $P \cap R$ *urejene* po teži. Brez težave lahko dobimo časovno zahtevnost $O(k \log n + \log n)$ za posamezno poizvedbo. Brez težave lahko dobimo časovno zahtevnost $O(k \log n + \log n)$ za posamezno poizvedbo. Dokaži, da se lahko to naredi v času $O(k \log \log n + \log n)$ za posamezno poizvedbo. Nasvet: najprej dokaži, da lahko dobimo urejen seznam števil iz t urejenih seznamov števil v času $O(m \log t)$, kjer je m število elementov v vseh seznamih skupaj.

Možen je tudi čas $O(k + \log n)$ za posamezno poizvedbo, vendar rabi hude ideje.

5.11. Opiši psevdokodo za vstavljenje in brisanje točke iz kd-drevesa. Ni nujno paziti na poravnavanje drevesa.

5.12. kd-drevesa lahko uporabljamo tudi za iskanje točk, ki so v drugih tipih območij, na primer v krogih ali trikotnikih.

(a) Zapiši psevdokodo za iskanje točk, ki so v vhodnem trikotniku.

(b) Dokaži, da je čas odgovora v najslabejšem primeru linearen.

(c) Naredi (a) in (b) še za kroge.

5.13. Videli smo, da 2-dimenzionalna območna drevesa porabijo $O(n \log n)$ prostora. Porabo želimo zmanjšjati s povečanjem časa odgovorjanja na poizvedbe.

(a) Predpostavimo, da imajo 'associated' strukture samo vozlišča na sodnih nivojih. Kakšna je adaptacija algoritma za pravilno odgovarjanje na poizvedbe.

(b) Koliko prostora porabi ideja iz točke (a)?

(c) Predpostavimo, da imajo 'associated' strukture samo vozlišča na nivojih $0, \lfloor \frac{1}{j} \log n \rfloor, \lfloor \frac{2}{j} \log n \rfloor, \ldots,$ kjer je $j > 2$ konstanta. Koliko prostora porabi takšna podatkovna struktura in koliko traja odgovor na posamezno poizvedbo?

5.14. Podana je množica n točk P v ravnini. Vsaka točka $p \in P$ ima težo $w_p \in \mathbb{R}$. Za vsako podmnožico $Q \subseteq P$ definiramo $w(Q) = \sum_{p \in Q} w_p$ in $w'(Q) = \max_{p \in Q} w_p$. Opiši podatkovno strukturo za hrambo P, ki vrne $w(P \cap R)$ in $w'(P \cap R)$ za vhodni pravokotnik R (poizvedba) v času $O(\log^2 n)$. Podatkovna struktura sme porabiti $O(n \log n)$ prostora, in mora biti konstruirana v času $O(n \log n)$.

5.15. Podana je množica n intervalov S na premici $\mathbb{R}$. Opiši algoritem, ki v času $O(n \log n)$ najde moč množice $\{(I, J) \in S^2 \mid I \subset J \text{ ali } J \subset I\}$.

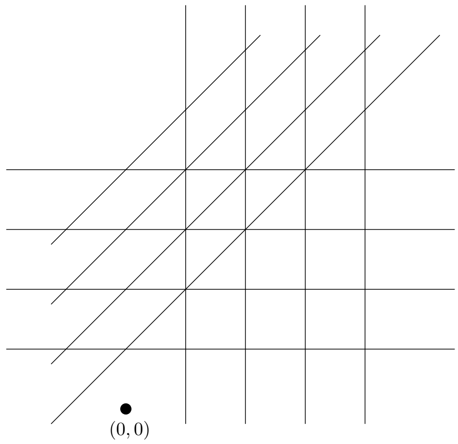
5.16. Danih je n intervalov na $\mathbb{R}$, $I = \{[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n]\}$. V času $\tilde{O}(n)$ predprocesiraj podatke, da boš znal nato v času $\tilde{O}(1)$ za poljuben dani interval $[p, q]$ povedati, v koliko intervalih iz I je vsebovan. Opiši oba postopka, tako predprocesiranje kot odgovarjanje na poizvedbe.

6 DCEL, razporeditve in dualnost

6.1. Naj bo H množica n premic v ravnini, ki se sekajo v isti točki p, in naj bo H' množica m premic v ravnini, ki se sekajo v drugi točki $p' \neq p$. Premice iz H ne potekajo skozi p', premice iz H' pa ne potekajo skozi p.

(a) Koliko oglišč, robov in lic ima razporeditev $\mathcal{A}(H)$?

(b) Koliko oglišč, robov in lic ima razporeditev $\mathcal{A}(H \cup H')$?



Slika 6.28: Delna slika za $n = 4$ v vaji 6.3.

(c) Naj bo R množica n ravnin v $\mathbb{R}^3$. Vsaka ravnina iz R vsebuje točko (0,0,0) in nobena trojka ravnin se ne seka na premici. Koliko oglišč, robov, lic in celic ima razporeditev $\mathcal{A}(R)$?

6.2. Naj bosta L_h množica n vodoravnih premic v ravnini in L_v množica n navpičnih premic v ravnini. Naj bo L_p množica n premic v ravnini, ki se sekajo v isti točki p. Nobena premica iz L_p ni vodoravna ali navpična in nobena izmed vodoravnih ali navpičnih premic ne vsebuje točke p. Poleg tega se nobene tri premice $\ell_h \in L_h, \ell_v \in L_v, \ell_p \in L_p$ ne sekajo v isti točki.

(a) Koliko oglišč, robov in lic ima razporeditev $\mathcal{A}(L_h \cup L_v \cup L_p)$?

(b) Opišite, kako izgledajo točke, dualne premicam $L_h \cup L_p$.

6.3. Za vsak i naj bo ℓ_i premica $\{(x, y) \in \mathbb{R}^2 \mid x = i\}$, naj bo v_i premica $\{(x, y) \in \mathbb{R}^2 \mid y = i\}$ in naj bo d_i premica $\{(x, y) \in \mathbb{R}^2 \mid y = x + i\}$. Za naravno število n, naj bo

$$L = \{\ell_1, \ell_2, \ldots, \ell_n\} \cup \{v_1, v_2, \ldots, v_n\} \cup \{d_1, d_2, \ldots, d_n\}.$$

Glej sliko 6.28. Koliko oglišč, robov in lic ima razporeditev $\mathcal{A}(L)$?

6.4. Naj bo $S = \{C_1, \ldots, C_n\}$ množica n krožnic polmera 1. Predpostavimo, da so v splošnem položaju: nobene 3 krožnice se ne sekajo v isti točki in nobeni 2 krožnici se ne dotikata. Lice razporeditve $\mathcal{A}(S)$ je maksimalna povezana množica v $\mathbb{R}^2 \setminus (\bigcup_i C_i)$. Dokaži naslednje trditve.

(a) Če seka vsaka krožnica iz S največ k krožnic iz S, potem ima $\mathcal{A}(S)$ največ $O(n(k+1))$ lic.

(b) Če za vsako točko p velja $\{C_i \mid C_i \text{ vsebuje } p\}| \leq k$, potem ima $\mathcal{A}(S)$ največ $O(nk)$ lic.

6.5. Podan je DCEL povezane subdivzije v ravnini. Vsako lice subdivizije, ki ni zunanjo lice, je *konveksno*. Opiši (učinkovito) psevdokodo za naslednje naloge.

(a) Podano je lice f. Navedi oglišča lica f.

(b) Podano je lice f. Določi, ali je f zunanje lice.

(c) Podano je oglišče v. Navedi lica, ki imajo v na robu.

(d) Podano je oglišče v. Najdi oglišče DCELa, ki ima najmanjšo x-koordinato.

(e) Podano je lice f. Navedi lica, ki imajo vsaj eno oglišče skupno s f.

(f) Podana je daljica $\overline{pq}$, ki seka robove DCELa, in lice f, ki vsebuje p. Predpostavimo, da $\overline{pq}$ ne vsebuje oglišč DCELa. Ažuriraj DCEL, da bo imel novo subdivizijo s $\overline{pq}$. (Glej sliko 6.30.)

6.6. Kakšna je množica premic, ki so dualne točkam trikotnika $\triangle pqr$?

6.7. Naj bo P množica n točk v ravnini. Točka $x \in \mathbb{R}^2$, ne nujno v P, je *osrednja točka* za P, če ima naslednjo lastnost: vsaka polravnina, ki vsebuje x, vsebuje vsaj $n/3$ točk iz P.

- Kaj to pomeni v dualnem prostoru?
- Opiši algoritem, ki najde osrednjo točko, če obstaja.

6.8. Naj bo S podana množica n daljic v ravnini.

(a) Opiši algoritem, ki v času $O(n^2)$ odloči, ali obstaja premica, ki seka vsako daljico v S.

(b) Opiši algoritem, ki vrne premico, ki seka največje število daljic v S.

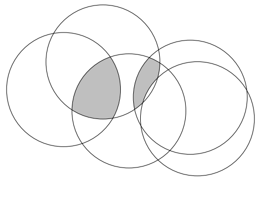
6.9. Naj bo P množica n točk v ravnini in naj bo $w \notin P$ točka. Za vsako premico ℓ naj bo $\phi(\ell) = \max_{p \in P} d(p, \ell)$, kjer je $d(p, \ell)$ razdalja med p in ℓ. Opiši algoritem, ki vrne premico ℓ^*, ki vsebuje w in maksimizira $\phi(\ell)$.

6.10. Naj bo L množica n premic in naj bo P množica n točk v ravnini. Zanima nas, ali obstajata točka $p \in P$ in premica $\ell \in L$ tako, da je $p \in \ell$. Opiši algoritem, ki rešuje ta problem grobo v času $O(n^{3/2})$. (Namig: naredimo skupine, ki vsebujejo $\sqrt{n}$ premic.)

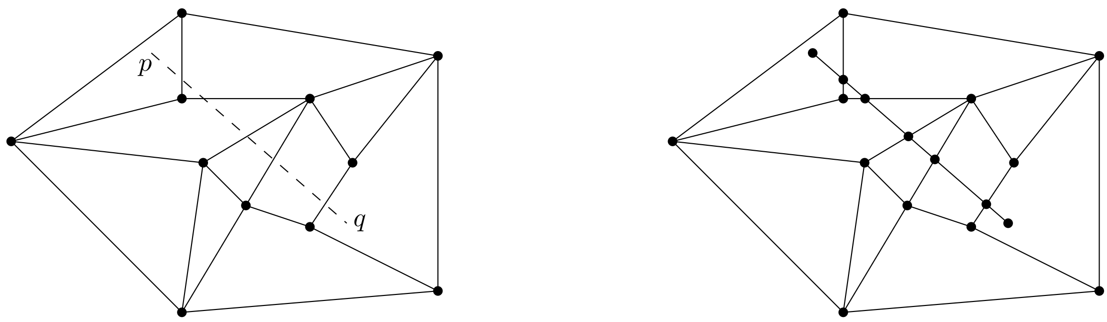
Kako hitro lahko rešujemo ta problem za n premic in k točk?

6.11. Naj bo P množica n točk v ravini. Opiši podatkovno strukturo, ki ga lahko konstruiramo v času $O(n^2)$ in v času $O(\log n)$ določi, koliko točk iz P so nad vhodno premico (poizvedba).

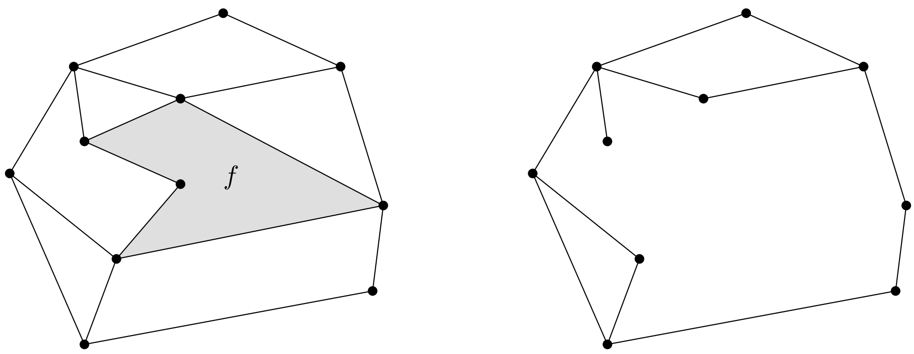
6.12. Naj bosta P in Q množica točk v ravnini. P in Q imata isto moč. Predpostavimo, da je $P \cap Q = \emptyset$ in da nobene 3 točke iz $P \cup Q$ niso kolinearne. Za P in Q velja naslednja pogoja:



Slika 6.29: Množica krožnic S polmera 1 in dve lici razporeditve $\mathcal{A}(S)$.



Slika 6.30: Ažuriranje po vstavljanju daljice v DCEL.



Slika 6.31: Ažuriranje po zbrisanju robov lica v DCEL.

- vsaka točka iz P je oglišče konveksne lupine $CH(P \cup Q)$;
- vsaka točka iz Q je oglišče konveksne lupine $CH(Q)$.

Opiši, kako izgledajo in kakšne lastnosti imajo premice dualne točkam $P \cup Q$.

6.13. Podana je množica ravnin H v splošnem položaju v $\mathbb{R}^3$: ne obstajajo 4 ravnine, ki se sekajo v isti točki. *Celica* razporeditve $\mathcal{A}(H)$ je maksimalna povezana podmnožica v $\mathbb{R}^3 \setminus \bigcup H$. *Lica, robovi, oglišča* razporeditve $\mathcal{A}(H)$ so lica, robovi, oglišča razporeditve $\mathcal{A}(\{h' \cap h \mid h' \in H\})$ na vsakem $h \in H$.

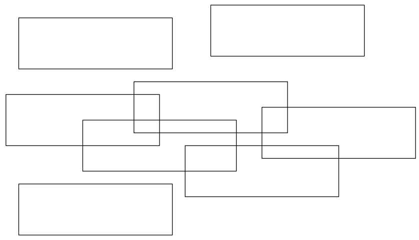
Koliko celic, lic, robov, oglišč ima $\mathcal{A}(H)$?

6.14. Naj bo L množica premic v ravnini. Oglišče v razporeditve $\mathcal{A}(L)$ je na nivoju 0, če nobena premica iz L ni pod v. Če je vsako oglišče iz $\mathcal{A}(L)$ na nivoju 0, kakšna je L?

6.15. Podana sta subdivizija ravnine v DCEL in lice f te subdivizije. Predpostavimo, da je subdivizija taka, da je vsako lice definirano s ciklom brez ponovljenih povezav. Želimo zbrisati vse povezave, ki definirajo f, to je $D = \{e \mid e$ je povezava na robu lica $f\}$. Predpostavimo, da subdivizija ostane povezana, ko zbrišemo D. Glej sliko 6.31. Zapiši psevdokodo, ki zbriše D in ažurira DCEL. Kakšna je časovna zahtevnost algoritma?

6.16. Opiši drugo definicijo dualnosti, kjer je vodoravna razdalja invariantna. Dokaži, da ne obstaja dualnost med točkami in premicami, pri kateri sta tako vodoravna kot navpična razdalja invariantni.

6.17. Kakšna je povezava med diagramom normal konveksnega večkotnika P in zgornjo ali spodnjo ovojnico premic, ki so dualne ogliščem večkotnika P?



Slika 6.32: Primer množice $\mathbb{D}$ v vaji 6.24.

6.18. Naj bodo $P_1, P_2, \ldots, P_k$ konveksni večkotniki v ravnini. Če obstaja premica, ki seka vse večkotnike $P_1, \ldots, P_k$, kakšna je situacija v dualnem prostoru?

6.19. Naj bosta R, B množici rdečih oziroma modrih točk v ravnini. Opiši algoritem, ki vrne polravnino, ki vsebuje vse modre točke in čim manj rdečih točk.

6.20. Dana je množica n modrih in n rdečih točk v ravnini. Opiši algoritem, ki v času $\tilde{O}(n^2)$ ugotovi, ali obstaja pas širine 1, ki vsebuje vse modre in nobene rdeče točke.

6.21. Zbodno število triangulacije večkotnika je največje število diagonal, ki jih seka poljubna premica v ravnini. Opiši algoritem, ki izračuna zbodno število podane triangulacije večkotnika v času $O(n^2)$, kjer je n število oglišč večkotnika.

6.22. Pravimo, da je večkotnik *k-konveksen*, če poljubna premica njegov rob seka največ k-krat. 2-konveksnost je torej ekvivalentna "navadni" konveksnosti. Opiši algoritem s časovno zahtevnostjo $O(n^2)$, ki za podani n-kotnik P in število k ugotovi, ali je P k-konveksen.

6.23. Naj bo P množica n točk v ravnini. Za premico ℓ v ravnini, ki ni navpična, naj bo $\phi(\ell) = \max\{d_{vert}(p, \ell) \mid p \in P\}$, kjer je $d_{vert}(p, \ell)$ navpična razdalja med točko p in premico ℓ. Naj bo ℓ^* premica, ki minimizira vrednost $\phi(\ell)$. Opiši algoritem, ki v času $O(n \log n)$ vrne premico ℓ^*.

6.24. Naj bo $\mathbb{D}$ množica n pravokotnikov, ki so translacije pravokotnika $[0, 3] \times [0, 1]$. Ni nujno, da so pravokotniki disjunktni. Glej sliko 6.32 za primer. Zanima nas, ali obstaja premica s pozitivnim naklonom, ki seka vsak pravokotnik iz $\mathbb{D}$.

(a) Kakšen je problem v dualnem prostoru?

(b) Opiši algoritem, ki rešuje tak problem v linearnem času.

(c) Opiši algoritem, ki v linearnem času odloči, ali obstaja premica z naklonom med 10 in 15, ki seka vsak pravokotnik iz $\mathbb{D}$.

(d) Opiši algoritem, ki v linearnem času odloči, ali obstaja premica, ki seka vsak pravokotnik iz $\mathbb{D}$. Pazi, naklon je zdaj povsem neomejen.

Naklon premice $\{(x, y) \in \mathbb{R}^2 \mid y = ax + b\}$ definiramo za potrebe te naloge kar kot njen smerni koeficient a.

6.25. Naj bo R množica n rdečih daljic v ravnini, in naj bo M množica n modrih daljic v ravnini. Premica ℓ je *akrobatska* za R in M, če seka $CH(R)$ in velja

$$|\{s \in R \mid \ell \cap s \neq \emptyset\}| = |\{s \in M \mid \ell \cap s \neq \emptyset\}|.$$

Opiši algoritem, ki v času $O(n^2)$ odloči, ali obstaja akrobatska premica za R in M.

6.26. Naj bo S množica n točk v ravnini.

(a) Opiši podatkovno strukturo, ki lahko v času $O(\log n)$ odloči, ali je neka točka iz P nad podano premico ℓ. (Premica ℓ nikoli ni navpična.)

(b) Opiši podatkovno strukturo, ki lahko v času $O(\log^2 n)$ odloči, ali je neka točka iz P nad podano daljico e. (Točka p je nad daljico e, ko je p nad premico, ki vsebuje e, in navpična premica skozi p seka e.)

(Namig: Za (b) lahko uporabiš večnivojsko podatkovno strukturo, kjer je prvi nivo za x-koordinate, drugi nivo pa za ugotavljanje položaja glede na premico, ki vsebuje e.)

7 Linearno programiranje

7.1. Večkotnik P je *zvezden*, če obstaja točka $p \in P$, ki vidi cel večkotnik. Opiši algoritem, ki v linearnem času odloči, ali je podan večkotnik P zvezden. Če je podan večkotnik P zvezden, ali lahko najdemo v linearnem času točko, ki vidi cel P? Pa množico točk, ki vidijo cel večkotnik?

7.2. Naj bo S podana množica navpičnih daljic v ravnini. Opiši algoritem, ki v linearnem času odloči, ali obstaja premica, ki seka vsako daljico v S.

7.3. Naj bodo $c_1, \ldots, c_n$ avtomobili, ki se premikajo po premici $\mathbb{R}$. Položaj avta c_i v času t je podan s formulo $x_i(t)$, ki je afina, t.j. oblike $x_i(t) = a_i t + b_i$; parametra a_i, b_i sta znana. Diameter avtomobilov $\Delta(t)$ v času t je definiran kot

$$\Delta(t) := \max\{x_1(t), x_2(t), \ldots, x_n(t)\} \; - \; \min\{x_1(t), x_2(t), \ldots, x_n(t)\}.$$

Opiši algoritem, ki najde minimalni diameter, ko je $t \in [0, 1000]$.
Opiši algoritem, ki najde maksimalni diameter, ko je $t \in [0, 1000]$.
Kdaj obstaja maksimalni diameter, ko je $t \geq 0$?

7.4. Naj bo S množica daljic v ravnini. Opiši algoritem, ki v linearnem času odloči, ali obstaja vertikalna daljica dolžine 1, ki seka vsako daljico iz S.

7.5. Naj bosta R in B množici rdečih oziroma modrih točk v ravnini. Opiši algoritme, ki rešujejo naslednje probleme v linearnem času:

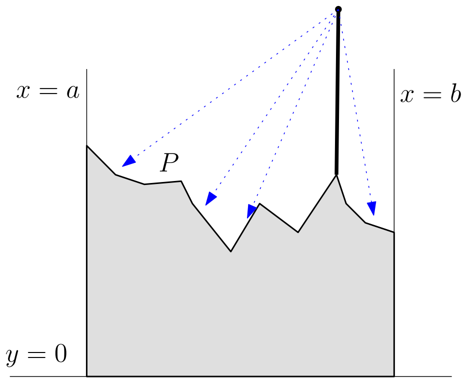
(a) Najti premico, če obstaja, ki loči R in B.

(b) (težje) Najti krog, če obstaja, ki ima R noter in B zunaj.

7.6. Naj bo P monotona poligonalna pot, ki ima n oglišč. Naj bosta a in b minimalna in maksimalna x-koordinata oglišč poti P. Pot P si lahko predstavljamo kot relief terena; glej sliko 7.33. *Stolp* je navpična daljica, ki ima bazo v poti P. Stolp vidi pot P, če je vsaka točka poti P vidna z vrha stolpa.

(a) Opiši algoritem, ki najde stolp, ki vidi P in minimizira y-koordinato vrha.

(b) Opiši algoritem, ki najde najkrajši stolp, ki vidi pot P.

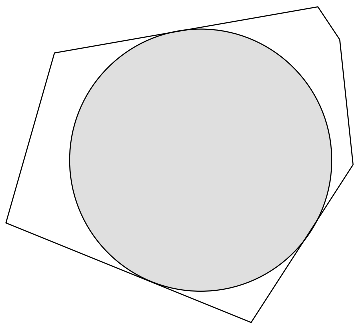


Slika 7.33: Pot P in stolp, ki vidi pot P.

7.7. Naj bo P konveksen večkotnik, ki je podan s tabelo oglišč $p[1, \ldots, n]$, naštetih v smeri urinega kazalca.

(a) Opiši algoritem, ki odloči, ali obstaja krog polmera r znotraj večkotnika P.

(b) Opiši algoritem, ki najde največji krog znotraj večkotnika P. (Glej sliko 7.34.)



Slika 7.34: Zgled za vajo 7.7.

7.8. *Obroček* v ravnini je množica točk, ki imajo razdaljo med a in b od neke točke (središče). Glej sliko 7.35. Podana je množica točk P v ravnini. Opiši algoritem, ki najde v linearnem času obroček, ki vsebuje P in ima najmanjšo ploščino.

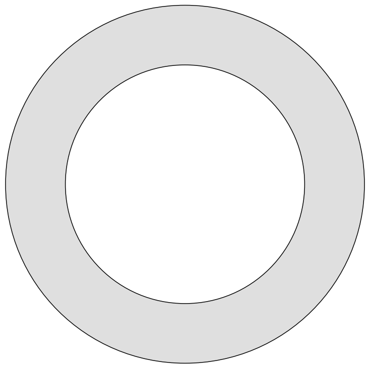
7.9. Naj bo P množica točk v ravnini in naj bo $q \notin P$ točka v ravnini.

(a) Z uporabo linearnega programiranja opiši algoritem, ki v linearnem času odloči, ali je točka q znotraj $CH(P)$.

(b) Opiši algoritem, ki v linearnem času določi interval I z naslednjo lastnostjo: $t \in I$ natanko tedaj, ko je točka $(t, 0)$ znotraj $CH(P)$.

7.10. Podane so konveksni večkotniki $P_1, P_2, \ldots, P_k$ v ravnini. Vsi večkotniki skupaj imajo n oglišč. Opiši algoritem, ki v linearnem času odloči, ali obstaja navpična daljica dolžine 1, ki je znotraj vsakega večkotnika.

7.11. Živimo na omejenem intervalu $I = [a, b] \subset \mathbb{R}$. Na intervalu I je n mest, ki jih označimo $M_1, M_2, \ldots, M_n$. Mesto M_i je na položaju $p_i \in I$ in ima parameter $w_i > 0$,



Slika 7.35: Obroček.

ki meri, kako pomembno je mesto. Večja vrednost w_i pomeni, da je mesto M_i bolj pomembno.

Iščemo prostor za novo bolnišnico, ki seveda mora biti znotraj intervala I. Če postavimo novo bolnišnico na položaj $x \in I$, potem lahko definiramo razdaljo od mesta M_i do bolnišnice kot $w_i \cdot |x - p_i|$. Zanima nas položaj bolnišnice, ki minimizira razdaljo do najbolj oddaljenega mesta. Na primer, recimo da je $I = [0, 4]$, mesto M_1 je na položaju $p_1 = 0$ s parametrom $w_1 = 3$, mesto M_2 pa je na položaju $p_2 = 3$ s parametrom $w_2 = 1$. Če postavimo bolnišnico na položaj $x = 1$, je razdalja od M_1 enaka 3 in razdalja od M_2 enaka 2. V tem položaju je M_1 bolj oddaljen od bolnišnice kot M_2.

Opiši algoritem, ki najde najboljši položaj nove bolnišnice v času $O(n)$.

7.12. V ravnini ležita dva konveksna večkotnika z nepraznim presekom. Opiši algoritem, ki v linearnem času izračuna najmanjšo razdaljo, za katero moramo premakniti P v podani smeri v, da bosta P in Q disjunktna.

7.13. Naj bo M konveksen sedemkotnik v ravnini z oglišči $p_1, p_2, \ldots, p_7$, naštetimi v smeri urinega kazalca. naštetimi v smeri urinega kazalca. Translacija večkotnika M za vektor (t_x, t_y) je množica točk $\{(x + t_x, y + t_y) \mid (x, y) \in M\}$. Za $\alpha \in \mathbb{R}$, $\alpha > 0$, z M^α označimo sedemkotnik, ki ga dobimo iz M z raztegom ravnine za parameter α. Torej je $M^\alpha = \{(\alpha \cdot x, \alpha \cdot y) \mid (x, y) \in M\}$.

Naj bo Q množica n točk v ravnini.

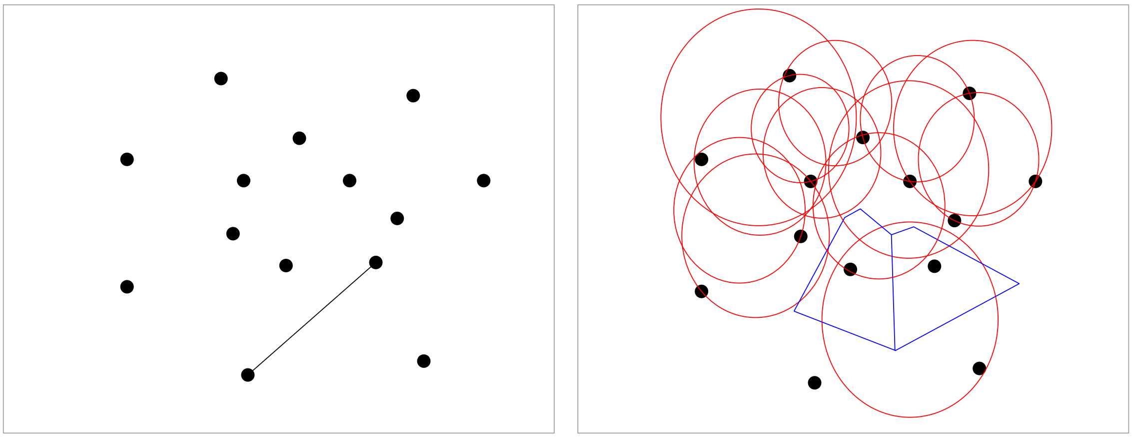
(a) Opiši algoritem, ki v linearnem času odloči, ali obstaja translacija večkotnika M, ki vsebuje Q.

(b) Opiši algoritem, ki v linearnem času poišče najmanjši α, za katerega obstaja neka translacija večkotnika M^α, ki vsebuje Q.

7.14. Po nočnem nebu se premika n kometov. Astronom Arne bi jih rad posnel s svojim fotoaparatom, in sicer s čim večjo povečavo. Če nekoliko poenostavimo, lahko nebo predstavimo z ravnino, komete s premočrtno premikajočimi se točkami, kos neba, ki ga zajame fotoaparat, pa s pravokotnikom s stranicami v razmerju 4:3. Velikost pravokotnika je odvisna od povečave, in to lahko Arne poljubno spreminja. Ravnino neba opremimo s koordinatnim sistemom tako, da je pravokotnik fotoaparata poravnan s koordinatnimi osmi; stojalo aparatu dopušča poljubno premikanje po ravnini neba, ne pa tudi sukanja. V trenutku, ko Arne začne opazovati nebo, je i-ti komet na položaju (x_i, y_i) in se premika s hitrostjo $(v_{x,i}, v_{y,i})$ enot na sekundo. Ob katerem času ima Arne priložnost narediti sliko vseh kometov hkrati z največjo možno povečavo? Opiši postopek, ki najde odgovor v času $O(n)$.

8 Voronojev diagram in Delaunayeva triangulacija

8.1. V sliki 8.36 vriši Delaunayevo triangulacijo podane množice točk.



Slika 8.36: Za vajo 8.1 Levo: točke in eno povezavo Delaunayeve triangulacije. Desno: Podatke

8.2. Konstruiraj Voronojev diagram točk $(0,0),(3,2),(1,5),(4,4),(5,8),(6,1)$ glede na razdaljo L_1. Zapiši koordinate točk, ki so pomembne v diagramu.

8.3. Konstruiraj Voronojev diagram točk $(0,1),(2,2),(4,5),(6,3),(1,9)$ glede na razdaljo L_∞. Zapiši koordinate točk, ki so pomembne v diagramu.

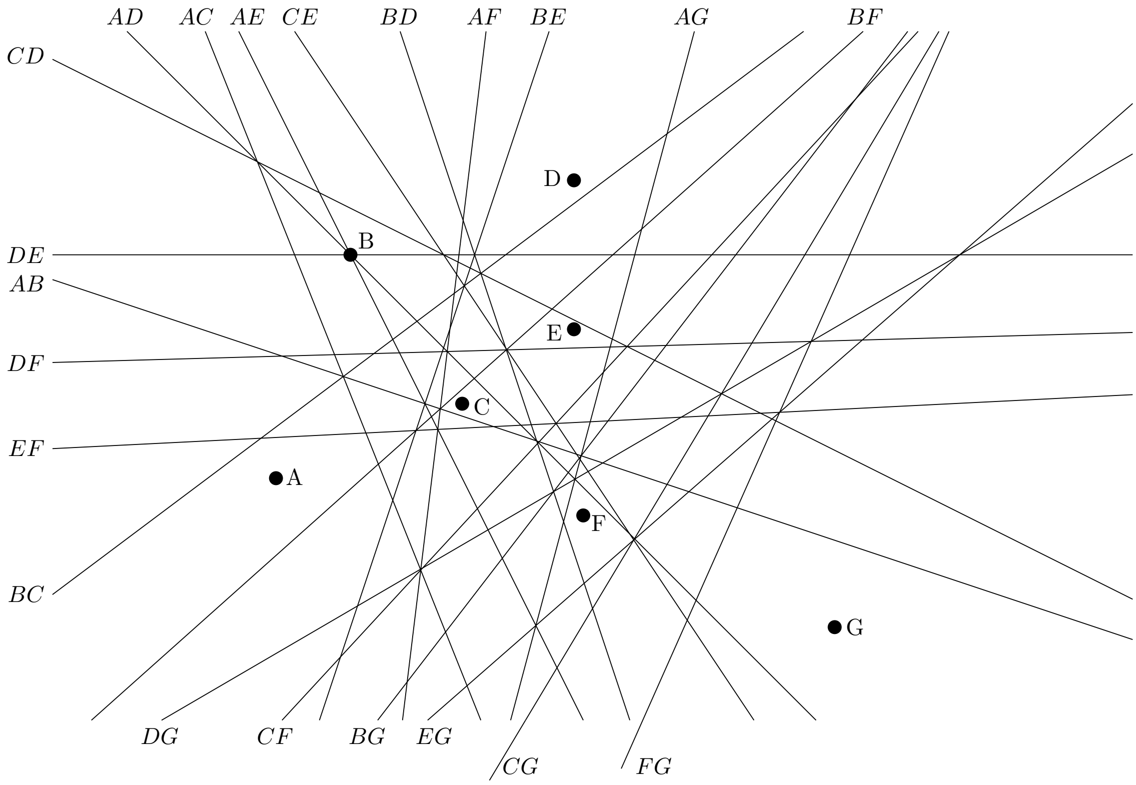
8.4. Množica sedmih točk $\mathcal{P}=\{A,B,C,D,E,F,G\}$ je vrisana na sliki 8.37. Na sliki je za vsak par točk $X,Y\in\mathcal{P}$, $X\neq Y$, narisana in ustrezno označena tudi simetrala daljice XY. Navedi povezave Delaunayeve triangulacije točk $\mathcal{P}$.

8.5. Za $i=1,\ldots,100$ izberi koordinate točke $p_i\in\mathbb{R}^2$ na tak način, da sta v Delaunayevi triangulaciji množice $\{p_1,p_2,\ldots,p_{100}\}$ najmanj 2 točki stopnje 99. Točke morajo biti v splošnem položaju: nobene tri točke niso kolinearne in nobene štiri točke niso kocirkularne. Natančno utemelji, da ima tvoja izbrana množica točk želene lastnosti.

8.6. Naj bo P množica n točk v ravnini. Najti želimo najbljžjo točko $q_p\in P$ za vsako točko $p\in P$, to je točko p_q, ki zadovolji $|pq_p|=\min\{|pq|\mid q\in P\setminus\{p\}\}$. Opiši algoritem, ki vrne q_p za vsako $p\in P$ v času $O(n\log n)$.

8.7. Naj bo P množica n točk v ravnini. Naj bo $K(P)$ poln evklidski graf na P; to je, P je množica vozlišč grafa $K(P)$ in za vsak par $p,q\in P$ obstaja povezava pq s težo $|pq|$. Naj bo T minimalno vpeto drevo grafa $K(P)$. Dokaži, da je T podgraf Delaunyeve triangulacije množice P in da lahko izračunamo T v času $O(n\log n)$.

8.8. Naj bo P množica točk v ravnini. Opiši algoritem, ki v $O(n\log n)$ času vrne največji krog D, ki ima svoje središče v $CH(P)$ in ne vsebuje nobene točke iz P.
Kaj se zgodi, če odstranimo pogoj, da mora biti središče kroga v $CH(P)$?



Slika 8.37: Za vajo 8.4.

8.9. Naj bo M množica n modrih točk v ravnini in naj bo R množica n rdečih točk v ravnini. Opiši algoritem, ki v času $O(n \log n)$ izračuna

$$\min\{d(m,r) \mid m \in M, r \in R\},$$

pri čemer je $d(\cdot,\cdot)$ evklidska razdalja.

8.10. Naj bo P množiča n točk v ravnini, ki jih vidimo kot "ovire". Naj bo $D(x)$ krog polmera 1 s središčem v točki $x \in \mathbb{R}^2$. Naj bosta s, t točki v $\mathbb{R}^2$ z lastnostjo $P \cap D(s) = P \cap D(t) = \emptyset$. Opiši algoritem, ki v času $O(n \log n)$ odloči, ali obstaja zvezna krivulja $x(t)$ z lastnostmi: $x(0) = s$ in $x(1) = t$ ter $D(x(t)) \cap P = \emptyset$ za vsak $t \in [0,1]$. Manj formalno, zanima nas, ali obstaja premik za krog polmera 1 od položaja s do položaja t, pri katerem se ne dotaknemo točk iz P. Nasvet: Voronojev diagram lahko pomaga.

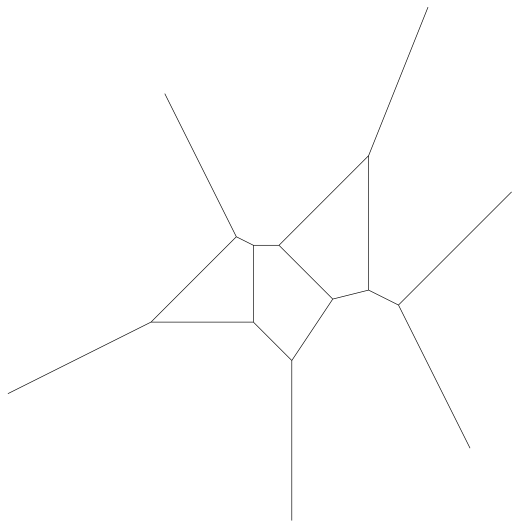
8.11. Naj bo P množica točk $\{p_i(i/n, 0, 0) \mid i = 1 \ldots n\} \cup \{q_j(0, j/n, 1) i = 1 \ldots n\}$ v $\mathbb{R}^3$. Dokaži, da je $\Omega(n^2)$ kombinatorična zahtevnost Voronojevega diagrama množice P.

8.12. Podan je Voronojev diagram $Vor(P)$ brez množice točk P, ki ga je definiral. Glej sliko 8.38. Ali lahko najdemo P v linearnem času?

8.13. Naj bo P množica n točk v ravnini. Zanima nas *najoddaljenejši Voronojev diagram*: Za vsako točko $p \in P$ naj bo

$$cel(p) = \{x \in \mathbb{R}^2 \mid d(p,x) \geq d(q,x) \text{ za vsako } q \in P \setminus \{p\}\}.$$

(a) Ali je $cel(p)$ konveksna množica? In v $\mathbb{R}^d$?



Slika 8.38: Zgled za vajo 8.12. Kje pa je P?

(b) Dokaži, da je $cel(p) \neq \emptyset$ natanko tedaj je p skrajna točka iz $CH(P)$.

(c) Naj bo E množica robov diagrama, to je, točke v ravnini, ki so v $cel(p)$ in $cel(p')$, $p \neq p'$. Dokaži, da E ne vsebuje sklenjene krivulje, to je, E je po strukturi drevo.

(d) Dokaži spodnjo mejo $\Omega(n \log n)$ za časovno zahtevnost računanja najoddaljenejšega Voronojevega diagrama.

(e) Opiši algoritem, ki izračuna najoddaljenejši Voronojev diagram v času $O(n \log n)$.

8.14. Naj bo P množica točk v ravnini. Zanima nas 2-Voronojev diagram $Vor_2(P)$. Za vsak par $p, q \in P$ naj bo

$$cel(\{p,q\}) = \{x \in \mathbb{R}^2 \mid d(x,p), d(x,q) \leq d(x,r) \text{ za vsako } r \in P \setminus \{p,q\}\}.$$

2-Voronojev diagram $Vor_2(P)$ je množica celic $cel(\{p,q\})$, ki niso prazne. Kakšne lastnosti imajo celice $cel(\{p,q\}) \in Vor_2(P)$? So povezane ali konveksne? Opiši kvadratičen algoritem, ki konstruira $Vor_2(P)$.

8.15. Spomni se na metriko L_1 v $\mathbb{R}^2$, kjer je razdalja med točkama (x,y) in (x',y') definirana kot $|x-x'| + |y-y'|$.

(a) Ali so celice Voronojevega diagrama pod metriko L_1 konveksne?

(b) Ali so povezane?

(c) Ali je kombinatorična zahtevnost Voronojevega diagrama pod metriko L_1 linearna?

(d) Konstruiraj Voronojev diagram točk $(0,0), (1,2), (5,5), (6,5), (0,9)$ pod metriko L_1.

(e) Opiši algoritem, ki konstruira Voronojev diagram pod metriko L_1.

8.16. Naj bo P množica n točk v ravnini. Koliko trikotnikov ima vsaka triangulacija točk P? (Namig: potreboval boš še en parameter.)

8.17. Naj bo P množica točk v ravnini. Njena *najkrajša* triangulacija je tista, ki minimizira vsoto dolžine robov. Dokaži, da najkrajša triangulacija ni nujno Delaunayeva triangulacija.

8.18. Naj bo $P = \{p_1, p_2, \ldots p_n\}$ množica n točk v ravnini. Predpostavimo naslednji splošni položaj: če je $\{p_i, p_j\} \neq \{p_{i'}, p_{j'}\}$, potem velja $d(p_i, p_j) \neq d(p_{i'}, p_{j'})$, kjer je $d(\cdot, \cdot)$ evklidska razdalja. Najmanšja razdalja d_{min} v P je

$$d_{min} = \min\{d(p_i, p_j) \mid p_i \neq p_j\}.$$

Drugo najmanjšo razdaljo $d_{2.min}$ v P definiramo kot

$$d_{2.min} = \min\left(\{d(p_i, p_j) \mid p_i \neq p_j\} \setminus \{d_{min}\}\right).$$

Opiši algoritem, ki v času $O(n \log n)$ vrne drugo najmanjšo razdaljo $d_{2.min}$.

8.19. Naj bo P množica n točk v ravnini. *Gabrielov graf* $G(P)$ za P ima P kot množico vozlišč in vključuje povezavo med točk p, q iz P natanko tedaj, ko krog s premerom pq ne vsebuje nobene točke iz $P \setminus \{p, q\}$.

(a) Pokaži, da je graf $G(P)$ podgraf Delaunayeve triangulacije za P.

(b) Pokaži, da je pq povezava v grafu $G(P)$ natanko tedaj, ko imata lici Voronojevega diagrama za p in q skupni rob e in daljica $\overline{pq}$ seka e.

(c) Opiši algoritem, ki vrne $G(P)$ v času $O(n \log n)$.

(d) Dokaži, da porabi izračunanje Gabrielova grafa najmanj $\Omega(n \log n)$ časa.

8.20. Naj bo P množica n točk v ravnini in naj bo $q \notin P$ dodatna točka znotraj $CH(P)$. Opiši algoritem, ki v času $O(n \log n)$ vrne največji krog D, ki pokrije q, vendar v notranjosti ne vsebuje nobene točke iz P. (Točke iz P lahko so na robu kroga.)

8.21. Naj bo P množica n točk v splošnem položaju v ravnini. V času $O(n \log n)$ poišči najmanjši krog, ki ima na svojem robu natanko tri točke iz P, v strogi notranjosti pa natanko eno točko s P.

8.22. Naj bo $\mathbb{D}$ množica n krogov polmera 1 v ravnini. Opiši podatkovno strukturo, ki hrani $\mathbb{D}$ in omogoča poizvedbe naslednje oblike: za podano točko p odloči, ali v $\mathbb{D}$ obstaja krog, ki vsebuje p. Čas gradnje podatkovne strukture mora biti $O(n \log n)$. Podatkovna struktura mora odgovoriti na poizvedbo v času $O(\log^{10} n)$.

Opiši drugo podatkovno strukturo za enak problem, kjer je $\mathbb{D}$ množica n pravokotnikov, ki so translacije pravokotnika $[0, 3] \times [0, 1]$.

8.23. Naj bo P množica n točk v ravnini. Iščemo podatkovno strukturo, ki hrani P in omogoča poizvedbe naslednje oblike: za podan krog D odloči, ali je množica $P \cap D$ prazna ali ne.

(a) Opiši takšno podatkovno strukturo, ki jo lahko konstruiramo v času $O(n \log n)$ in v času $O(\log n)$ odgovori na posamezno poizvedbo, če uporabljamo evklidsko razdaljo.

(b) Opiši podatkovno strukturo, ki jo lahko konstruiramo v času $O(n \log n)$ in v času $O(\log^2 n)$ odgovori na posamezno poizvedbo, če uporabljamo metriko L_1.

Pazi: (zaprt) krog D je množica točk, ki so od središča oddaljene za največ polmer kroga; pri tem je razdalja bodisi evklidska bodisi L_1.