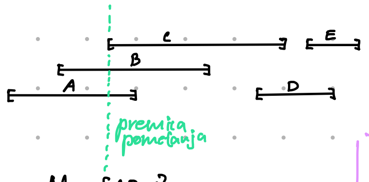
ALGORITMI POMETANJA

Presečišča intervalov



$M = \{A, B, C\}$

Stanje: intervali, ki jih seka premica pometanja

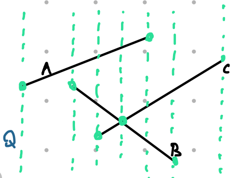
Dogodki: krajišča intervalov
- levo krajišče intervala I
  Dodaj presečišče $(I, J)\ \forall J \in M$
  Dodaj I v M
- desno krajišče
  Izbriši I iz M

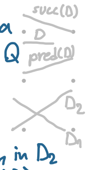
naštevanje:
$O(n \log n + k)$
└ št. intervalov
└ št. presečišč

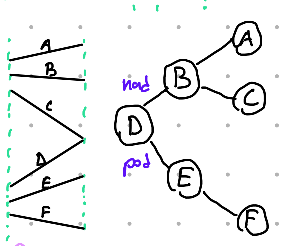
štetje, detekcija
$O(n \log n)$

Presečišča daljic

Dogodki: (prioritetna vrsta Q)
- levo krajišče daljice D
  - Dodaj D v stanje M
  - Dodaj morebitna presečišča med D in succ(D) ter D in pred(D) v Q
- desno krajišče daljice D
  - Dodaj morebitna presečišča med succ(D) in pred(D) v Q
  - Odstrani D iz stanja M
- presečišče med D_1 in D_2
  - Shrani presečišče
  - V stanju M zamenjaj D_1 in D_2
  - Dodaj morebitna presečišča med D_1 in pred(D_1) ter D_2 in succ(D_2) v Q







Stanje: (uravnoteženo binarno drevo)
- hrani daljice ki jih seka premica pometanja urejene po y-koordinati
- omogoča poizvedbi succ in pred

časovna zahtevnost
$O((n+k) \log n)$
└ št. premic
└ št. presečišč

prostorska zahtevnost
$O(n+k)$

KONVEKSNOST

$C \subseteq \mathbb{R}^d$ konveksna $\iff$ $\forall p \in C, q \in C\ \forall \lambda \in [0,1]: (1-\lambda)p + \lambda q \in C$

konveksna ovojnica $S \subseteq \mathbb{R}^d$ convex hull

$$CH(S) = \bigcap_{\substack{T \subseteq \mathbb{R}^d \\ T \text{ konv.} \\ S \subseteq T}} T \cong CH(S) = \{x \in \mathbb{R}^d \mid x = \sum_{i=1}^{n} \lambda_i s_i, \sum_{i=1}^{n} \lambda_i = 1, s_i \in S, \lambda_i > 0\}$$

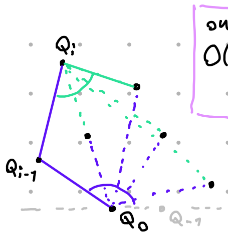
dovolj je $n = d+1$ catheodory

Izrek: Presek konveksnih množic je konveksen.

Algoritem zavijanje darila

Jarvis' march

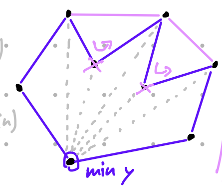
$Q_0 \leftarrow$ najnižja točka
$Q_{-1} = Q_0 + (1, 0)$
ponavljaj dokler ne prideš okrog
$Q_{i+1} \leftarrow$ točka p, ki minimizira $\measuredangle(Q_{i-1} - Q_i, p - Q_i)$



output-sensitive
$O(nh)$
└ $|S|$
└ $|CH(S)|$

Grahamov algoritem

- poiščemo najnižjo točko $O(n)$
- sestavimo večkotnik ↻ tako da točke uredimo po kotu $O(n \log n)$
- sprehodimo se po večkotniku in brišemo ne ekstremne točke kjer naredimo zasuk v levo $O(n)$

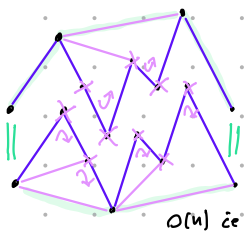


$O(n \log n)$

Inkrementalni algoritem

prirastni

- točke uredimo po x-koordinati
- posebej izračunamo zgornjo in spodnjo polovico
- dodajamo točke od leve proti desni. Če naredimo zavoj ↗ (zgoraj) / ↘ (spodaj) odstranimo točke do prejšnje maksimalne/minimalne



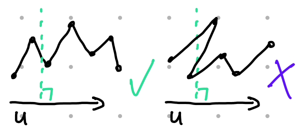
$O(n \log n)$

$O(n)$ če so že urejene

VEČKOTNIKI

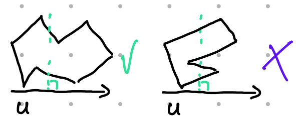
$\vec{u}$-monotona pot P

Presečišče med P in vsako premico pravokotno na $\vec{u}$ je lahko največ ena točka



$\vec{u}$-monoton večkotnik P

Presečišče med P in vsako premico $\perp \vec{u}$ je lahko največ ena daljica



Mejo monotonega večkotnika lahko razdelimo na dve monotoni poti

Razpoznavanje večkotnika

Ali se poligonska pot seka?

$O(n \log n)$ lahko tudi $O(n)$

Ploščina večkotnika

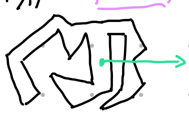
$$\sum_{i=1}^{n} \frac{1}{2}(x_{i+1} - x_i)(y_{i+1} + y_i)$$

$O(n)$

Ali je točka v večkotniku?

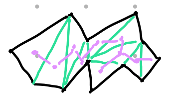
štejemo presečišča
liho → noter   sodo → zunaj

$O(n \log n)$



Triangulacija večkotnika

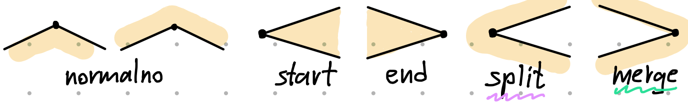
Razdelitev večkotnika na trikotnike. Oglišča trikotnikov morajo biti oglišča originalnega večkotnika.



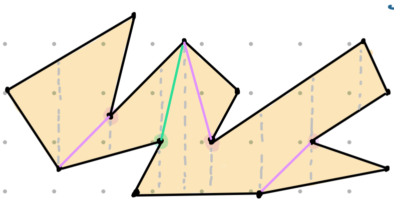
triangulacija
dualni graf (drevo stopnje največ 3)

Triangulacija večkotnika z n oglišč
- n oglišč
- n-3 diagonal
- n-2 trikotnikov

Razdelitev večkotnika na monotone večkotnike



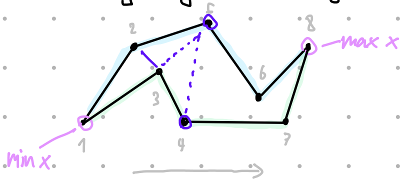
Večkotnik je (x)-monoton $\iff$ nima nobenega oglišča split ali merge



- večkotnik razdelimo na trapeze
- poiščemo split in merge oglišča in dodamo diagonalo do drugega oglišča v trapezu

$O(n \log n)$

Triangulacija monotonega večkotnika



Oglišča razvrstimo po x-koord. En konec elastike je na zgornji drugi pa na spodnji polovici. Na vsakem koraku dodamo diagonale (elastika).

$O(n)$

OSTALO

krovni izrek

$$T(n) = aT\left(\frac{n}{b}\right) + O(n^c) = \begin{cases} O(n^c) & a < b^c \\ O(n^c \log n) & a = b^c \\ O(n^{\log_b a}) & a > b^c \end{cases}$$

O-notacija

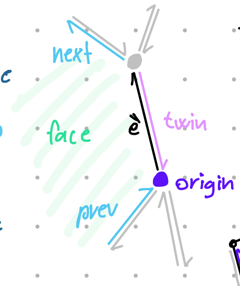
Notation	Description
$f(n) = o(g(n))$	f is dominated by g asymptotically $\forall k > 0\, \exists n_0\, \forall n > n_0: \lvert f(n)\rvert \le k\, g(n)$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$
$f(n) = O(g(n))$	$\lvert f\rvert$ is asymptotically bounded above by g $\exists k > 0\, \exists n_0\, \forall n > n_0: \lvert f(n)\rvert \le k\, g(n)$ $\limsup_{n\to\infty} \frac{f(n)}{g(n)} < \infty$
$f(n) = \Theta(g(n))$	f is asymptotically bounded by g both above and below $\exists k_1 > 0\, \exists k_2 > 0\, \exists n_0\, \forall n > n_0: k_1\, g(n) \le f(n) \le k_2\, g(n)$ $f(n) = O(g(n))$ and $g(n) = O(f(n))$
$f(n) \sim g(n)$	f is asymptotically equal to g $\forall \varepsilon > 0\, \exists n_0\, \forall n > n_0: \left\lvert \frac{f(n)}{g(n)} - 1 \right\rvert < \varepsilon$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1$
$f(n) = \Omega(g(n))$	f is bounded below by g asymptotically $\exists k > 0\, \exists n_0\, \forall n > n_0: f(n) \ge k\, g(n)$ $\liminf_{n\to\infty} \frac{f(n)}{g(n)} > 0$
$f(n) = \omega(g(n))$	f dominates g asymptotically $\forall k > 0\, \exists n_0\, \forall n > n_0: f(n) > k\, g(n)$ $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$

DCEL – doubly connected edge list

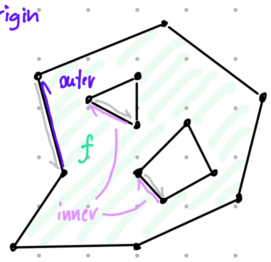
Hočemo opisati delitev ravnine na vozlišča, povezave in lica
(vertex, edge, face)

Half Edge
- origin : Vertex
- twin : Half Edge
  twin je vedno obrnjen v drugo smer
- face : Face
- next : Half Edge
- prev : Half Edge

Vertex
- coordinates
- edge : Half Edge
  ena izmed izhodnih povezav

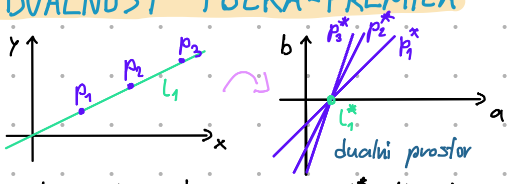


Face
- outer_component : HalfEdge
  ena izmed povezav na zunanjem ciklu
- inner_component : HalfEdge[]
  seznam povezav na notranjih ciklih



Vsa lica (razen zunanje) imajo outer_component.
inner_components ima lahko 0 ali več komponent.

DUALNOST TOČKA-PREMICA

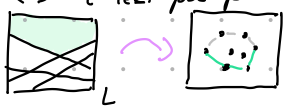


$l: y = l_a x - l_b$
$p: (p_x, p_y)$

$l^*: (l_a, l_b)$
$p^*: b = p_x a - p_y$

Lastnosti
- incidence preserving $p \in l \iff l^* \in p^*$
- $p_1, .., p_n$ kolinearne $\iff$ $p_1^*, .., p_n^*$ sekajo v isti točki
- p leži pod l $\iff$ l^* leži pod p^*

Zgornja ovojnica
upper envelope

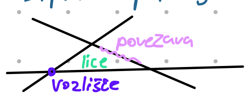


dualno spodnjemu delu $CH(L^*)$
oznaka: $LH(L^*)$

RAZPOREDITVE
ARRANGEMENTS

L ... množica n premic
↳ splošen položaj: ni vzporednic, ni presečišč 3 premic

$A(L)$



$|V| = \binom{n}{2}$
$|E| = n^2$
$|F| = \binom{n}{2} + n + 1$

zone(l) .... množica lic/celic, ki sekajo l

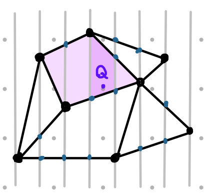
inkrementalna konstrukcija $A(L)$ kot DCEL $\boxed{O(n^2)}$

DOLOČANJE POLOŽAJA

Radi bi hitro ugotovili v katevi celici subdivizije ravnine leži dana točka Q.

subdivizije ravnine ↓ množica vozlišč povezav in lic ki se ne sekajo

▸ Naivna rešitev

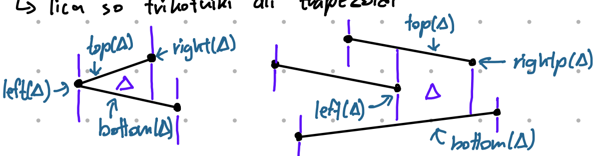


Q.x → bisekcija → pas

Q.y → pas → iskalno drevo → del lica → lice

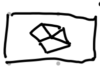
Poizvedba $O(\log n)$ konstrukcija $O(n^2 \log n)$
Prostor $O(n^2)$

▸ Vertical decomposition / trapezoidal decomposition
↳ lica so trikotniki ali trapezoidi



- Vsako lice hrani kazalce do svojih sosedov
- Vsaka povezava hrani svoji vozlišči in kazalec na originalno lice ki je nad njo.

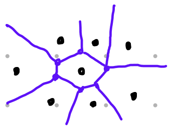
Vse točke zajamemo v agr. pravokotnik R



Inkrementalno dodajamo povezave v naključnem vrstnem redu in posodabljamo dekompozicijo T.
Sproti gradimo iskalno drevo D.

poizvedba $O(\log n)$ pričakovano
konstrukcija $O(n \log n)$
prostor $O(n)$

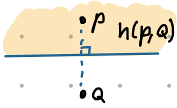
VORONOIJEV DIAGRAM



$P = \{p_1, \dots, p_n\}$ ... množica točk

Vor(P) ... Voronojev diagram (množica vozlišč, povezav in lic)

$V(p_i) = \bigcap_{j \neq i} h(p_i, p_j)$
↑ Voronojeva celica okoli p_i



$h(p,q)$ ... odprta polravnina

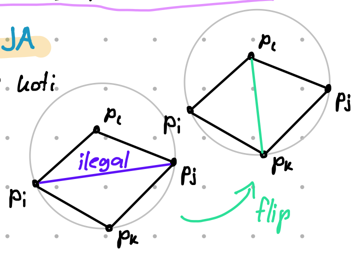
Izrek: Za $n \geq 3$ je v Voronojevem diagramu: $|V| \leq 2n-5$ $|E| \leq 3n-6$

Fortunes sweep line algorithm: $O(n \log n)$, $O(n)$ prostor

DELAUNAYEVA TRIANGULACIJA

Želimo triangulacijo s čim večjimi koti

Povezava $\overline{p_i p_j}$ je ilegalna
⇕
p_l leži v očrtani krožnici trikotnika $\Delta p_i p_j p_k$



Delaunayev graf je dualni graf (z poravnanimi povezavami) Voronoijevega diagrama.

Algoritmi:
- iterativno obračanje povezav
- plane sweep
- randomized incremental construction
- pretvorba iz voronoijevega diagrama

} $O(n \log n)$

Euclidian minimum spanning tree – EMST
↳ Graf ki povezuje vse točke iz P in ima minimalno skupno dolžino povezav.

Izrek: vsaka povezava v EMST je tudi v DT

Algoritem:
- izračunaj Delaunayevo triangulacijo DT(P)
- sortiraj povezave po dolžini
- dodajaj povezave dokler ne narediš cikla

$O(n \log n)$

OBMOČNA DREVESA
RANGE TREES

Imamo točke $P = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^d$

Želimo odgovoriti na vprašanje:

Katere točke so v $[a_1, b_1] \times [a_2, b_2] \times \dots \times [a_d, b_d]$?

	KD-tree (one tree)	range tree (one tree for first dimension, many next)
konstrukcija	$O(n \log n)$	$O(n \log n)$
poizvedba	$O(k + \sqrt{n})$	$O(k + \log^d n)$
prostor	$O(n)$	$O(n \log n)$

k output size

1D primer

- uravnoteženo binarno iskalno drevo
  z spremembo: ključe hranimo samo v listih
  vozlišča hranijo maksimum levega poddrevesa

poizvedba: $[6, 21]$