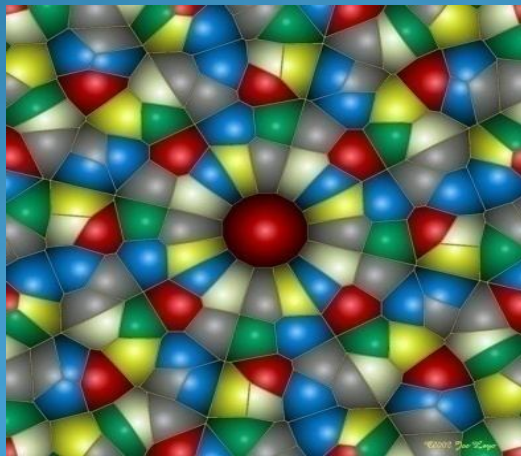




به نام خدا

فصل هفتم: دیاگرام ورونوی

محبوبه دریجانی



Voronoi Diagram
The postoffice problem

صورت مساله

- فرض کنید یک فروشگاه زنجیره‌ای می‌خواهد مکان‌هایی را برای احداث فروشگاه تعیین کند. برای بررسی مناسب بودن محل باید بررسی کرد که فروشگاه جدید چه اندازه می‌تواند در جلب مشتری موفق باشد. برای این کار باید بتوان ناحیه‌ای که تحت پوشش هر مرکز خرید قرار می‌گیرد را تعیین کرد.

صورت مساله

- فرض می‌کنیم
- قیمت هر کالا در همه جا یکسان است.
- هزینه به دست آوردن هر کالا برابر است با قیمت آن + هزینه حمل کالا از فروشگاه
- هزینه حمل کالا از فروشگاه به یک محل با فاصله آنها متناسب است.
- مشتریان سعی می‌کنند هزینه به دست آوردن کالا را مینیمم کنند.

صورت مساله

- با این فرض ها باید نقطه ای برای احداث فروشگاه تعیین شود که فاصله آن با همه فروشگاه های موجود زیاد باشد. به طور دقیق تر باید نقطه ای تعیین شود که فاصله آن تا نزدیکترین فروشگاه تا اندازه ممکن زیاد باشد.

صورت مساله - تعبیر هندسی

- مدل هندسی این مساله یک زیرتقسیم از صفحه به ناحیه‌هایی است که ساکنان هر ناحیه همه به یک مرکز خرید می‌روند. با فرض‌های قبل افراد مرکز خریدی را انتخاب می‌کنند که به آنها نزدیکتر است.
- پس برای هر مرکز خرید t می‌توان ناحیه‌ای را طوری تعیین کرد که نزدیکترین مرکز خرید همه نقاط آن t باشد.
- این ساختار **دیاگرام ورونوی** نام دارد.

دیاگرام ورونوی

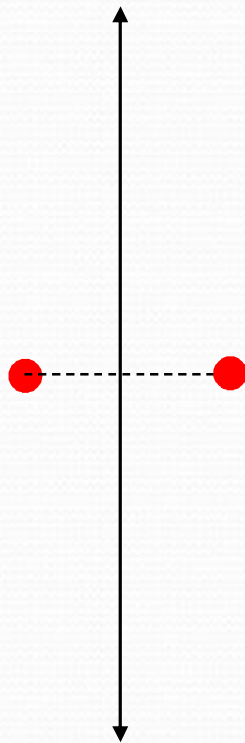


Georgy Voronoi
1868-1908

- مهمترین و پرکاربردترین ساختار بعد از convex hull
- همه اطلاعات مربوط به فاصله های مجموعه ای از نقطه ها (یا اشیای کلی تر)
- در سال 1850 توسط دیریکله مطرح شد و اولین مقاله در این زمینه در سال 1908 توسط ورونوی چاپ شد.

چند مثال

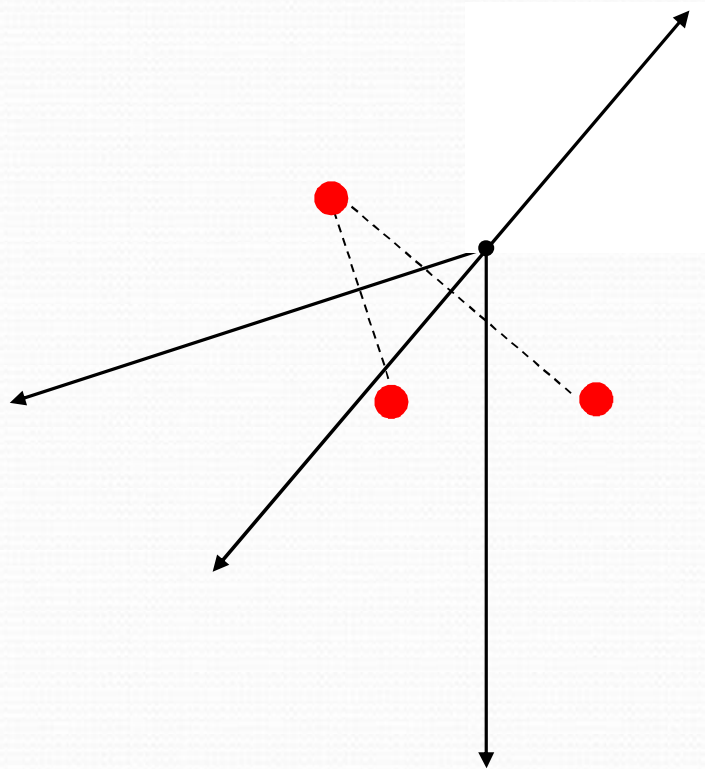
- دیاگرام ورونوی در حالت‌های ساده



- دیاگرام ورونوی یک نقطه تنها از همان نقطه تشکیل شده است.
- عمود منصف پاره خط متصل کننده دو نقطه دیاگرام ورونوی آنها است.

چند مثال

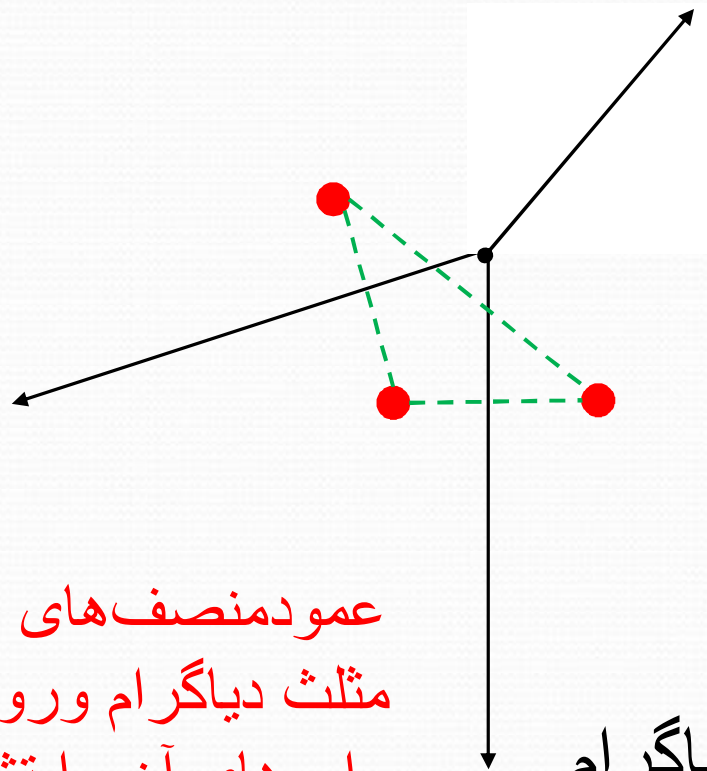
- دیاگرام ورونوی در حالت‌های ساده



- دیاگرام ورونوی یک نقطه تنها از همان نقطه تشکیل شده است.
- عمود منصف پاره خط متصل کننده دو نقطه دیاگرام ورونوی آنها است.

چند مثال

- دیاگرام ورونوی در حالت‌های ساده



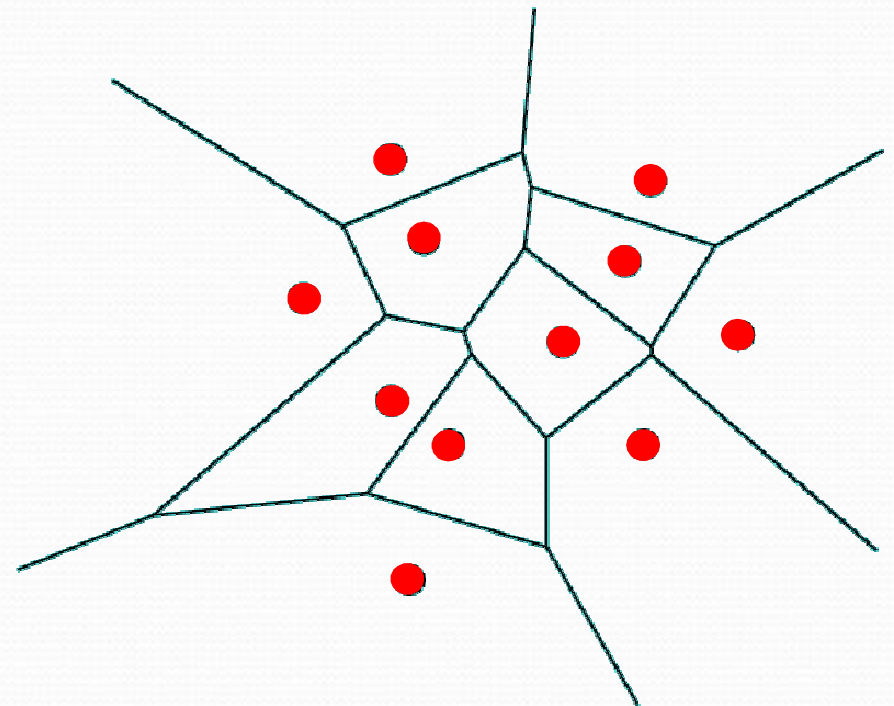
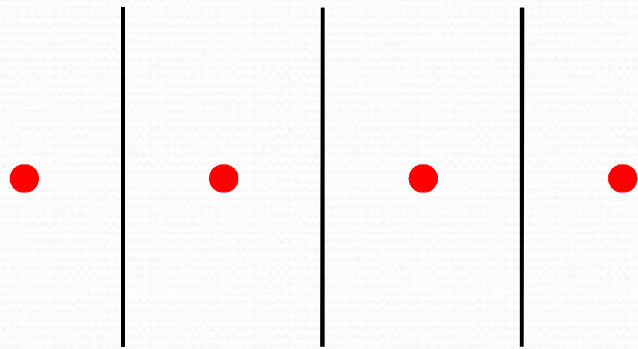
عمودمنصف‌های ضلع‌های
مثلث دیاگرام ورونوی
راس‌های آن را تشکیل
می‌دهند.

- دیاگرام ورونوی یک
نقطه تنها از همان نقطه
تشکیل شده است.

- عمودمنصف پاره‌خط
متصل کننده دو نقطه
دیاگرام ورونوی آنها
است.

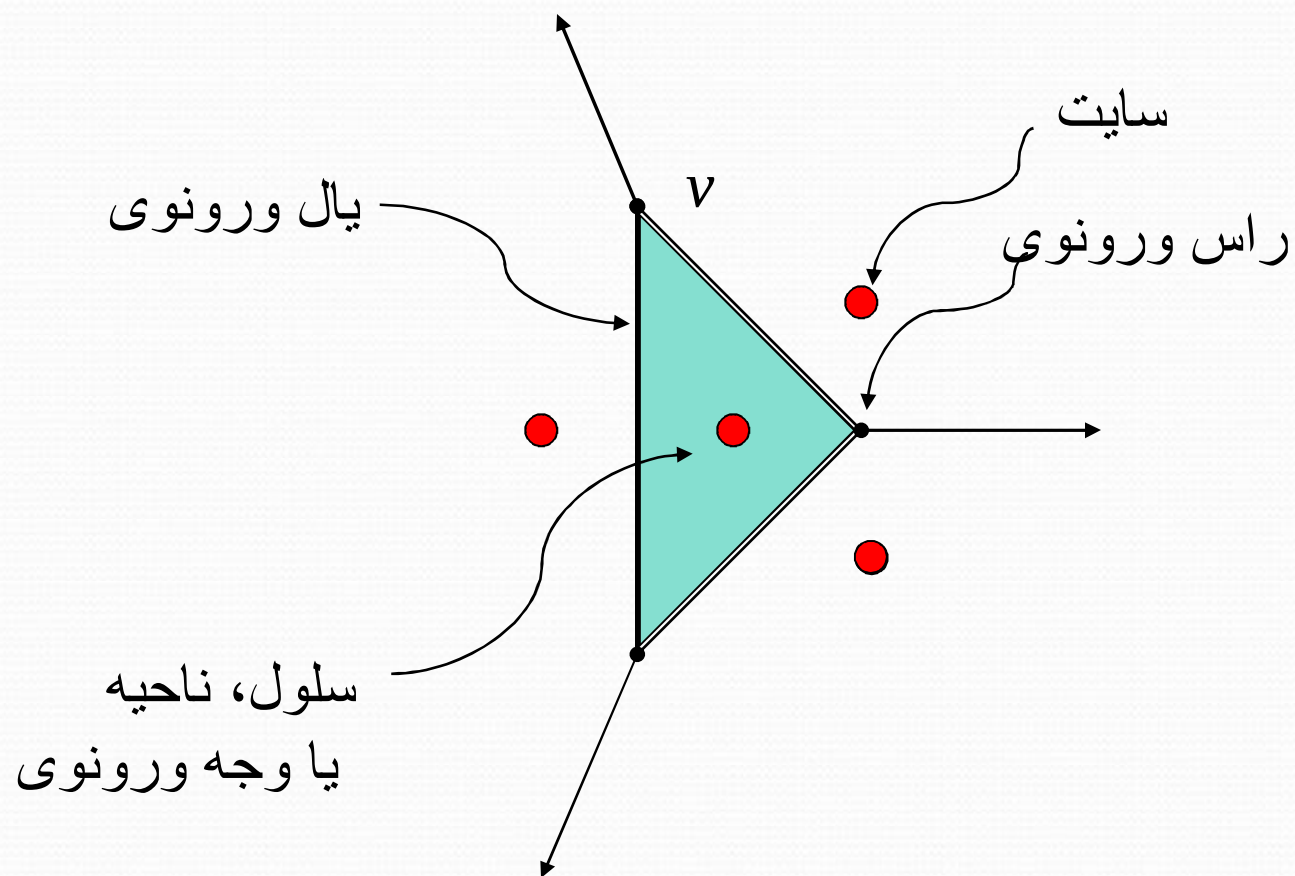
- برای سه نقطه غیر هم خط دیاگرام
ورونوی از سه نیم‌خط تشکیل شده
است.

چند مثال

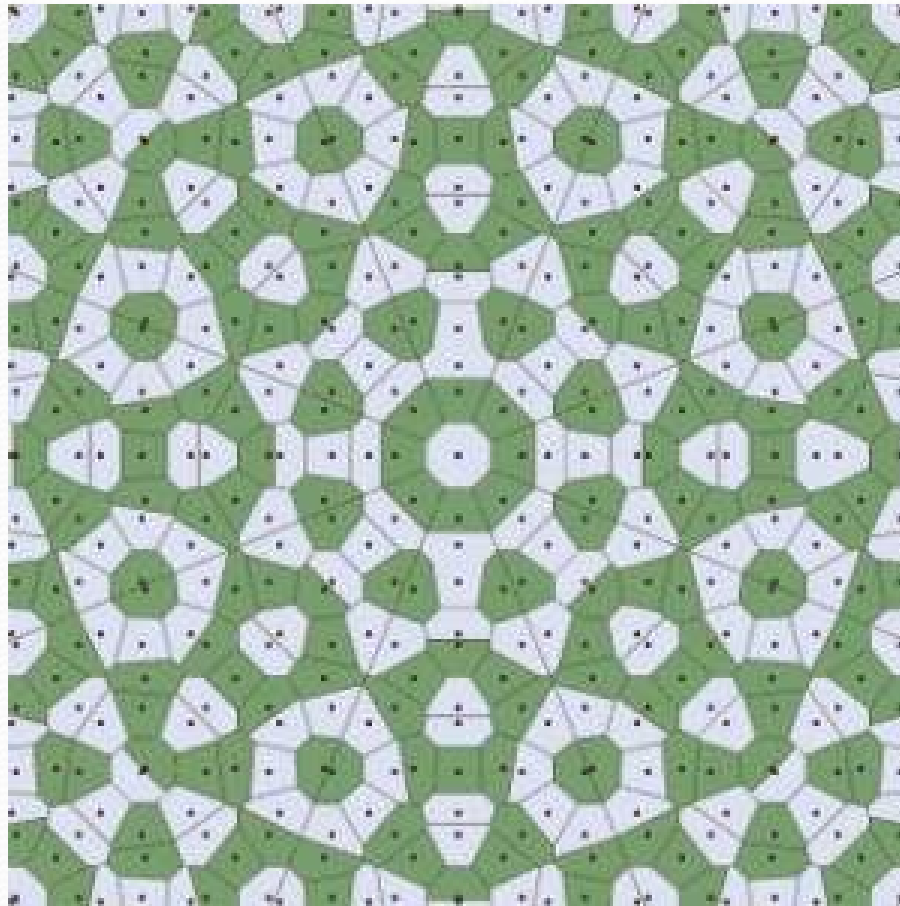


چند مثال

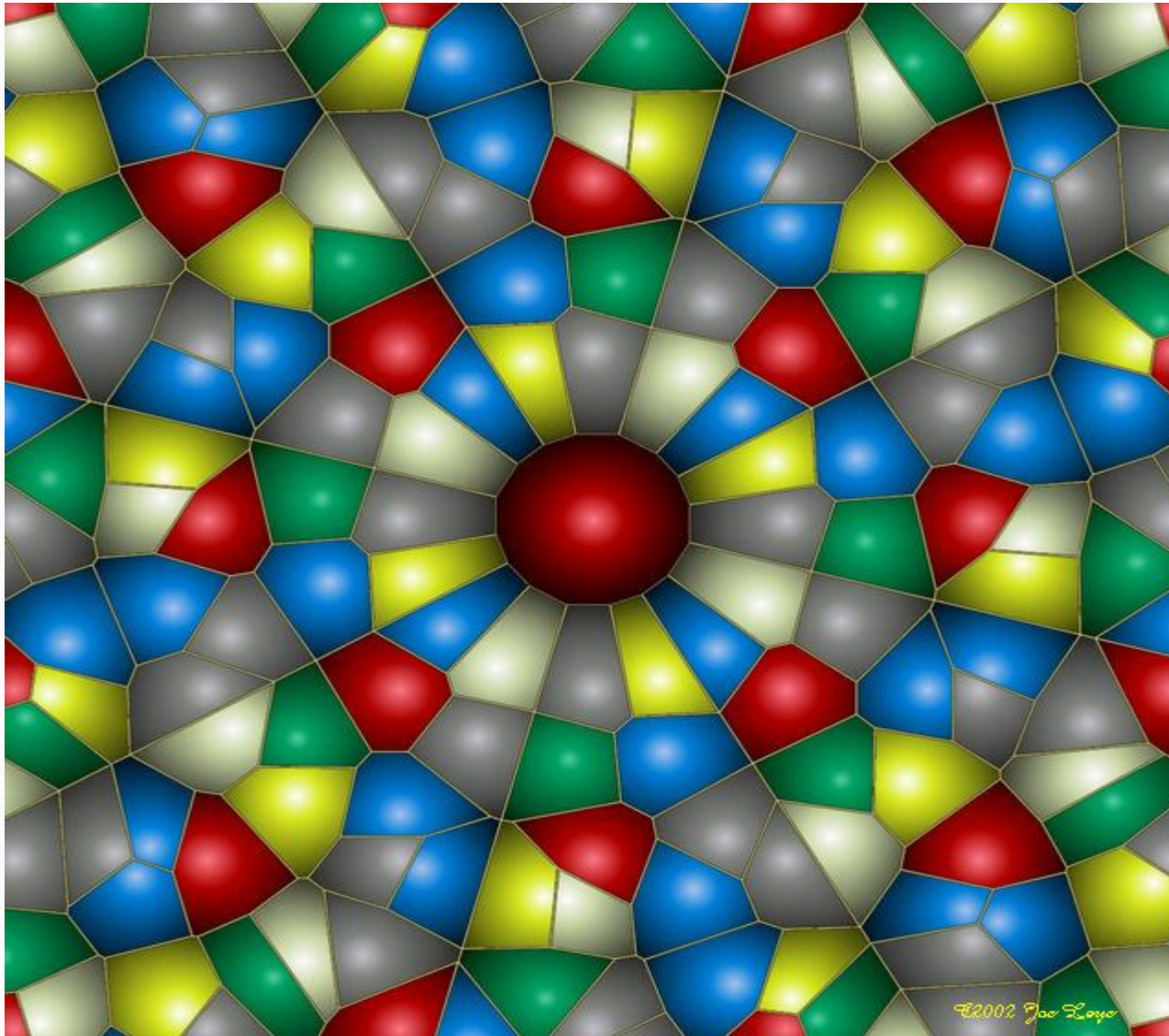
- اجزای دیاگرام ورونوی



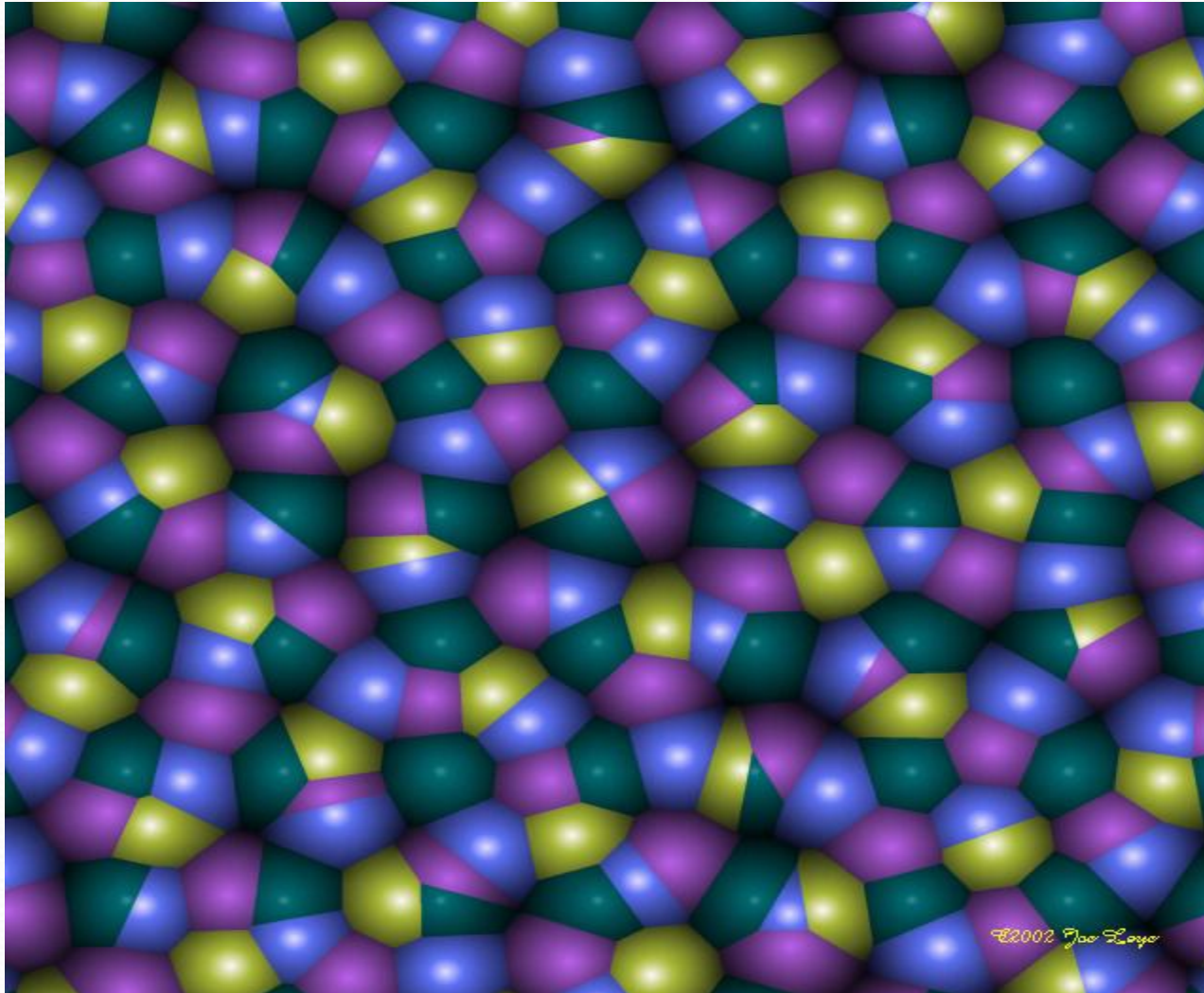
مثال



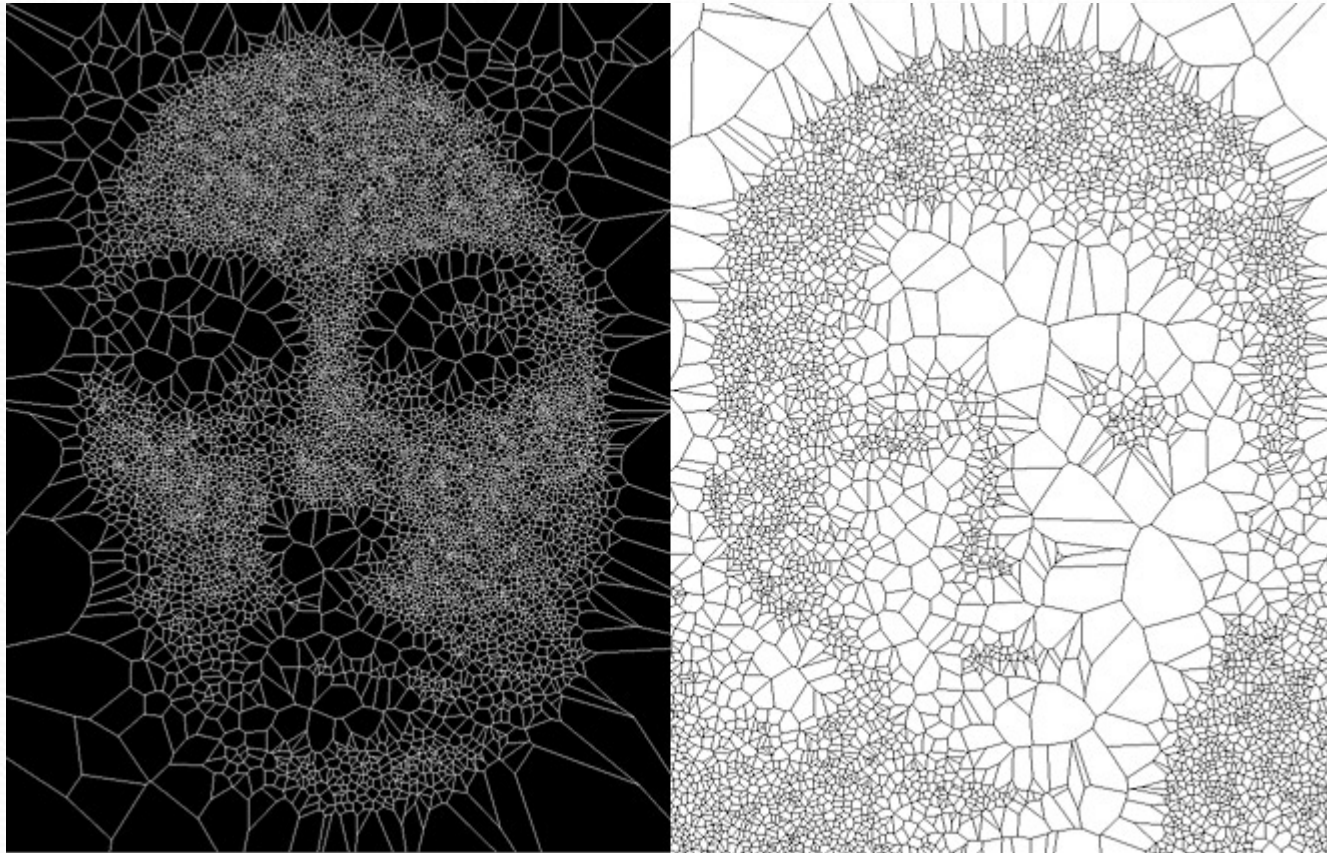
مثال



مثال

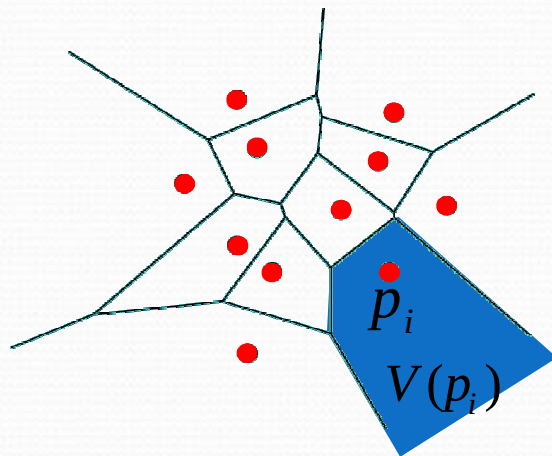


مثال



تعریف‌های اولیه

- مجموعه $P := \{p_1, p_2, \dots, p_n\}$ از نقاط در صفحه (سایت‌ها) یک زیرتقسیم از صفحه به n ناحیه (متناظر با سایت‌ها) به طوری که نقطه q در ناحیه متناظر با p_i است اگر و تنها اگر برای هر $p_j \in P$ که $j \neq i$ ، $\text{dist}(q, p_i) < \text{dist}(q, p_j)$.
دیاگرام ورونوی نام دارد.

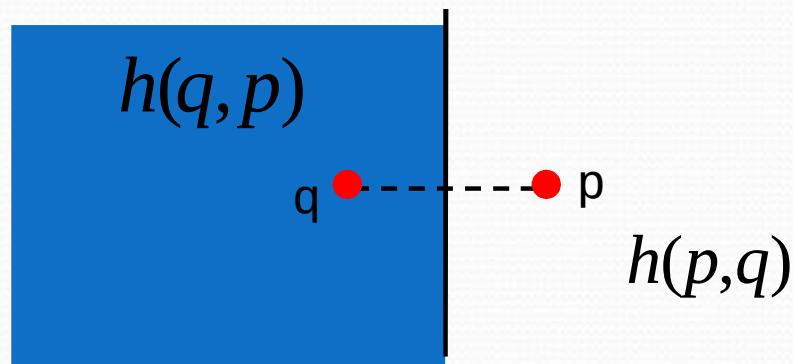


تعریف‌های اولیه

- فاصله اقلیدسی دو نقطه p و q به صورت زیر تعریف شده است:

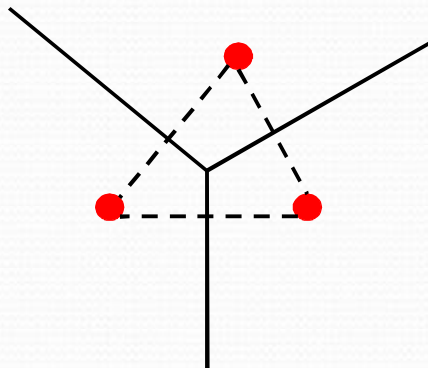
$$\text{dist}(p, q) := \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}.$$

- منصف دو نقطه p و q عمود منصف پاره‌خط pq است.



$r \in h(p, q)$ اگر و تنها اگر $\text{dist}(r, p) < \text{dist}(r, q)$

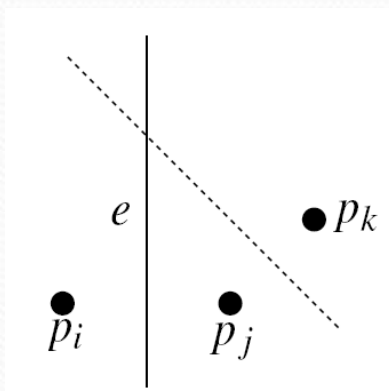
ساختار دیاگرام ورونوی



$$V(p_i) = \bigcap_{1 \leq j \leq n, i \neq j} h(p_i, p_j)$$

ساختار دیاگرام ورونوی

قضیه: فرض کنید P یک مجموعه از n نقطه (سایت) در صفحه باشد. اگر نقاط همخط باشند، دیاگرام ورونوی مجموعه‌ای از $n-1$ خط موازی است. در غیر این صورت دیاگرام ورونوی همبند است و هر ضلع آن یک پاره‌خط یا نیم‌خط است.
ایده اثبات:

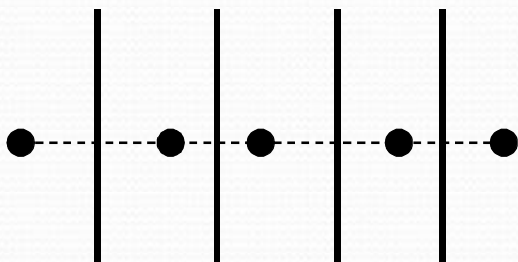


- ضلع‌ها قسمتی از عمودمنصف‌ها هستند.
- اگر یک ضلع خط باشد، مثلاً عمودمنصف p_i و p_j . نقطه‌ای مانند p_k هست که با این دو همخط نیست پس عمودمنصف آن این خط را قطع می‌کند.
- اگر دیاگرام ورونوی ناهمبند باشد یک ضلع آن خط است

ساختار دیاگرام ورونوی

قضیه: دیاگرام ورونوی با $n \geq 3$ سایت دارای حداکثر $2n - 5$ راس و $3n - 6$ ضلع است.

- ایده اثبات: اگر راس‌ها هم‌خط باشند دیاگرام ورونوی مجموعه‌ای از $n-1$ خط موازی است.



ساختار دیاگرام ورونوی

قضیه: دیاگرام ورونوی با $n \geq 3$ سایت دارای حداکثر $2n - 5$ راس و $3n - 6$ ضلع است.

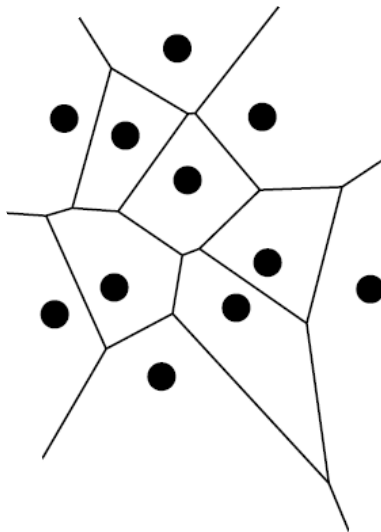
• ایده اثبات: اگر راس‌ها هم‌خط باشند دیاگرام ورونوی مجموعه‌ای از $n-1$ خط موازی است.

در غیر این صورت یک راس جدید v_∞ به دیاگرام اضافه می‌کنیم تا تبدیل به یک گراف مسطح شود.
با استفاده از فرمول اوایلر

$$(n_v + 1) - n_e + n = 2.$$

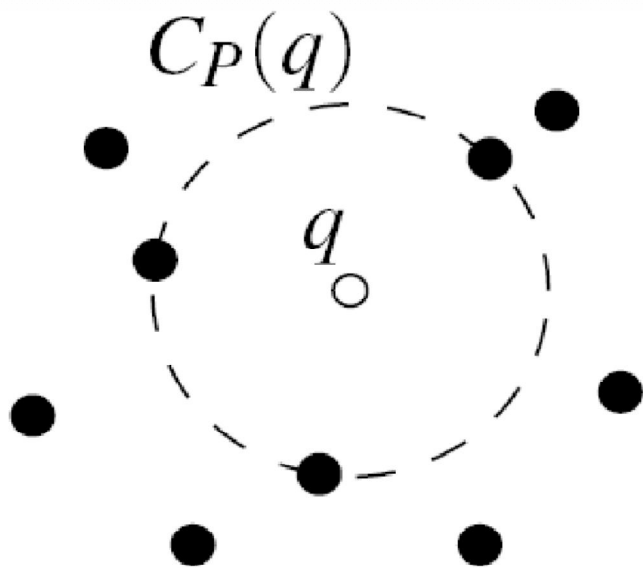
به علاوه هر ضلع روی مرز دو ناحیه است و هر راس حداقل سه یال مجاور است. پس $2n_e \geq 3(n_v + 1)$

در نتیجه $n_v \leq 2n - 5$ و $n_e \leq 3n - 6$



ساختار دیاگرام ورونوی

- بزرگترین دایره خالی q نسبت به مجموعه نقطه‌های P ، بزرگترین دایره با مرکز q است که هیچ نقطه P درون آن نیست.



ساختار دیاگرام ورونوی

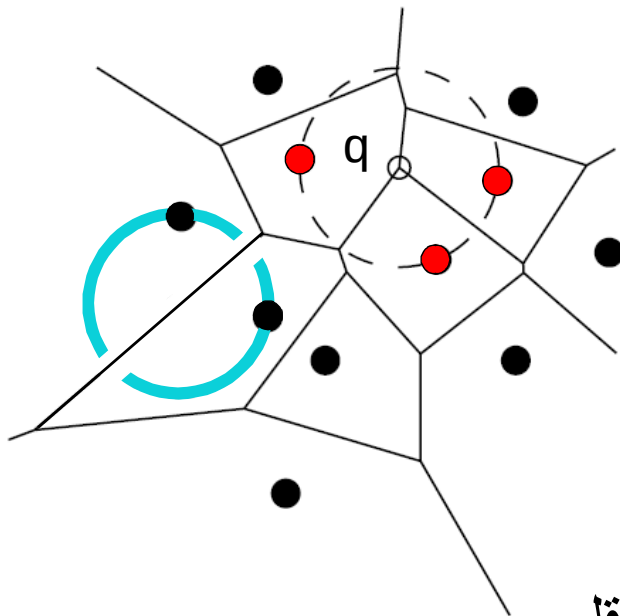
قضیه: برای دیاگرام ورونوی مجموعه P از نقاط خاصیت‌های زیربرقرار است.

(1) نقطه q راسی از دیاگرام ورونوی است اگر و تنها اگر حداقل سه راس P روی $C_P(q)$ باشند.

(2) عمودمنصف دو راس p_i و p_j ضلعی از $\text{Vor}(P)$ است اگر و تنها اگر راسی مانند q روی عمودمنصف باشد بطوریکه فقط p_i و p_j روی $C_P(q)$ باشند.

ساختار دیاگرام ورونوی

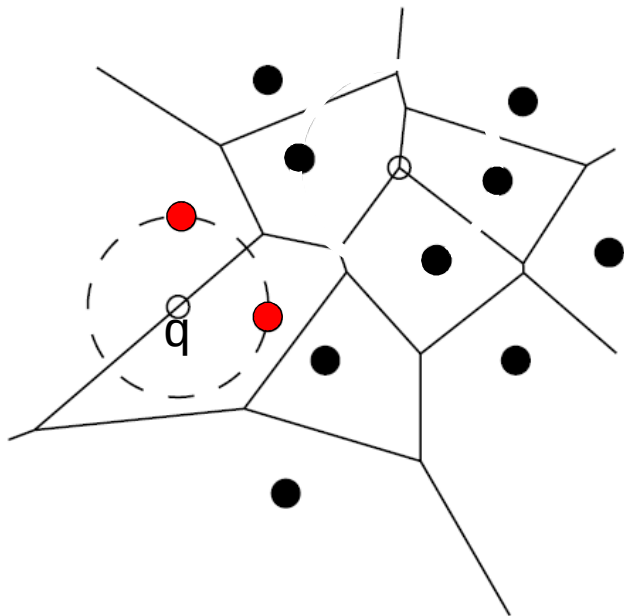
ایده اثبات:



(1) اگر q نقطه‌ای باشد که $C_P(q)$ از حداقل 3 سایت p_i و p_j و p_k عبور کند، آنگاه فاصله این سه نقطه تا q برابر است و هیچ سایتی به q نزدیکتر نیست. پس q در مرز ناحیه ورونوی هر سه سایت قرار دارد و یک راس ورونوی است.

اگر q راسی از دیاگرام ورونوی باشد، روی مرز حداقل سه ناحیه قرار دارد. پس سایت‌های این سه ناحیه فاصله مساوی تا q دارند و هیچ سایتی فاصله نزدیکتر تا q ندارد. در نتیجه حداقل سه راس P روی مرز $C_P(q)$ قرار دارند.

ساختار دیاگرام ورونوی



ایده اثبات:

(2) اگر q نقطه‌ای باشد که $C_p(q)$ از دقیقاً 2 سایت عبور کند. با استدلال قبل q یک راس یا یک ضلع از دیاگرام ورونوی است. اما q روی مرز دو ناحیه است. پس روی یک ضلع است.

برعکس، اگر قسمتی از عمودمنصف دو سایت p_i و p_j

یک ضلع از دیاگرام ورونوی باشد، بزرگترین دایره خالی در هر راس q روی این ضلع از دقیقاً دو سایت p_i و p_j عبور می‌کند.

ساختن دیاگرام ورونوی

یک الگوریتم اولیه:

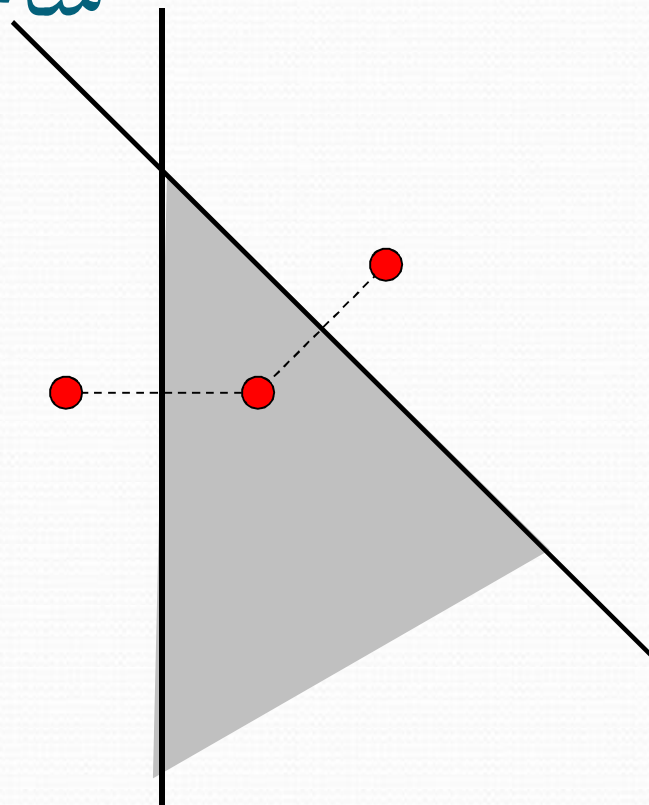
• با توجه به این برای هر سایت p_i ، $V(p_i) = \bigcap_{1 \leq j \leq n, i \neq j} h(p_i, p_j)$

می‌توان برای تعیین هر ناحیه ورونوی از اشتراک نیم‌صفحه‌ها استفاده کرد.

ساختار دیاگرام ورونوی

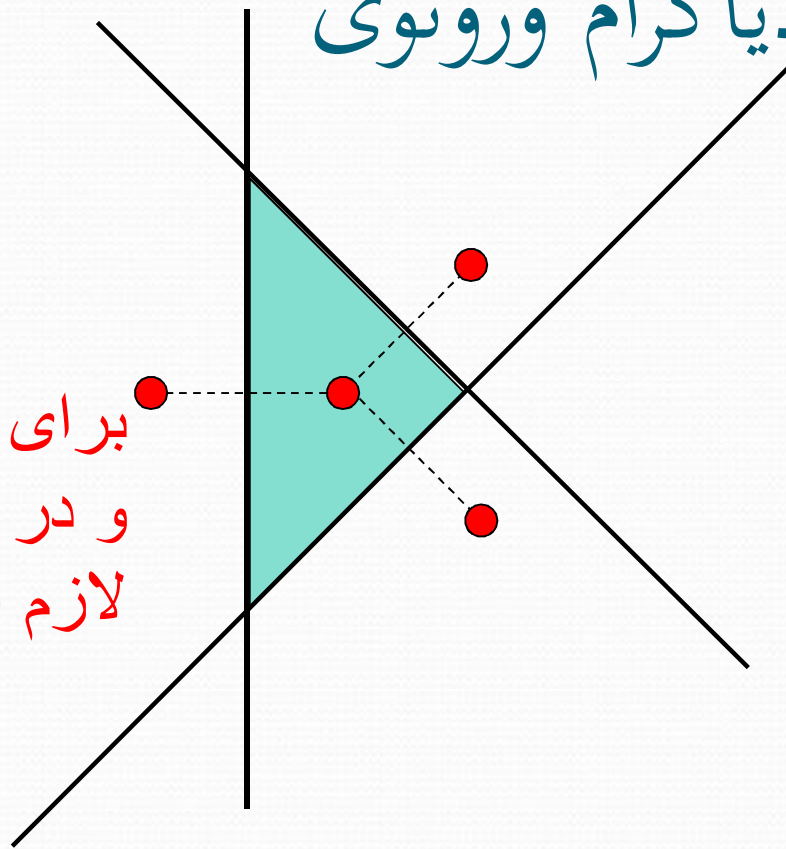


ساختار دیاگرام ورونوی



ساختار دیاگرام ورونوی

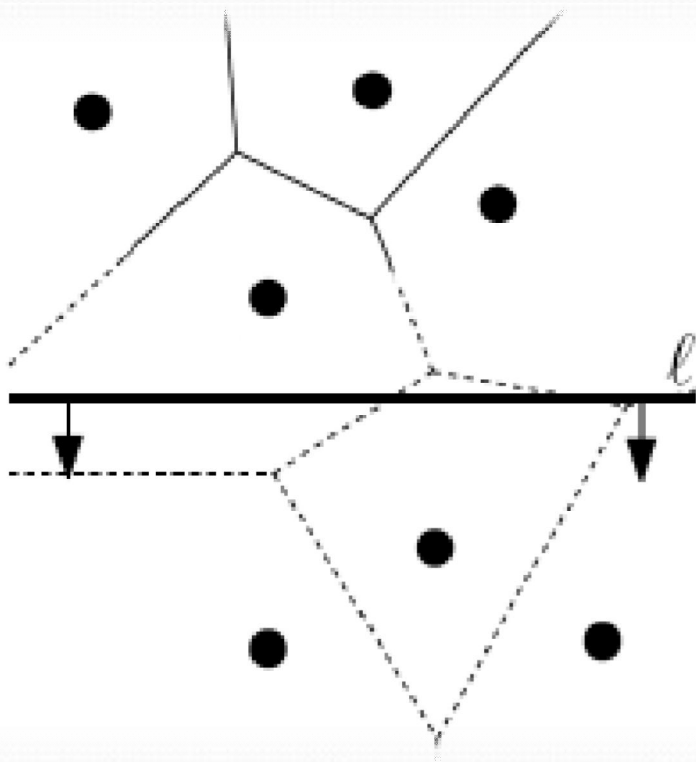
برای هر سایت زمان $O(n \log n)$
و در مجموع زمان $O(n^2 \log n)$
لازم است.



• آیا الگوریتم سریعتری وجود دارد؟

ساختن دیاگرام ورونوی

- ایده استفاده از الگوریتم *sweep line*



اشکال: دیاگرام ورونوی هم به سایت های بالای *sweep line* بستگی دارد و هم

به سایت های زیر خط *sweep line*

راه حل: در هر لحظه تنها قسمتی از

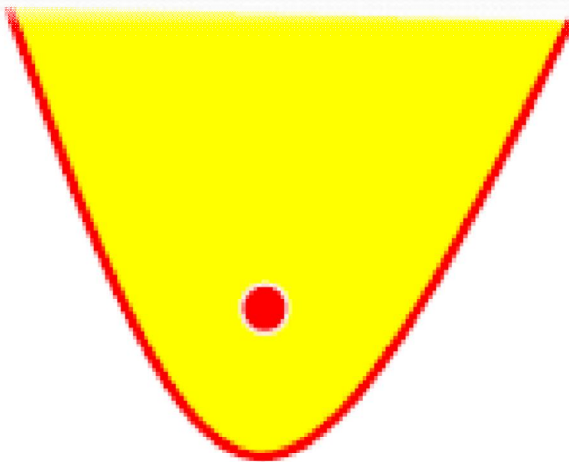
دیاگرام ورونوی که به سایت های زیر

sweep line بستگی ندارد را در نظر می گیریم. این قسمت با پایین رفتن

sweep line تغییر نمی کند

ساختن دیاگرام ورونوی

کدام قسمت از دیاگرام ورونوی با پایین رفتن / تغییر نمی کند؟



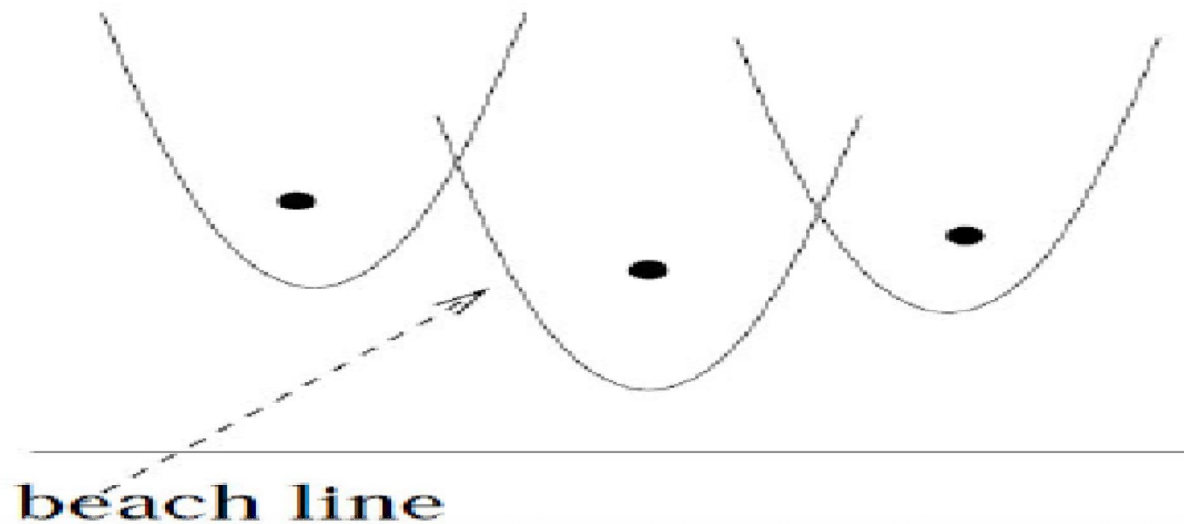
- قسمتی از صفحه که فاصله آن تا هر سایت زیر خط / از فاصله آن خط بزرگتر باشد.

- قسمتی از صفحه که فاصله آن تا خط / برابر با فاصله آن تا سایت P باشد.

مکان هندسی نقاطی از صفحه که فاصله آنها از یک نقطه و یک خط مساوی است یک سهمی است

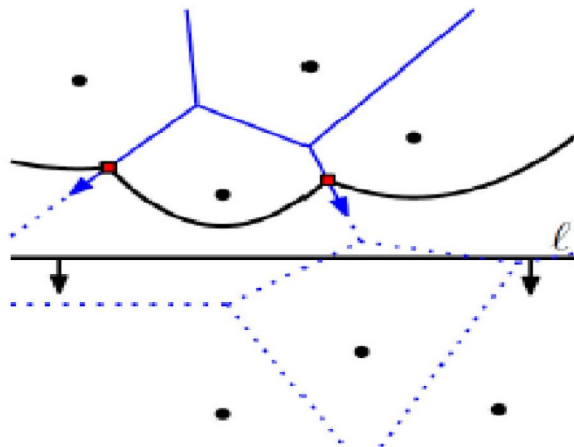
ساختن دیاگرام ورونوی

- هر سایت بالای / یک سهمی تعریف می کند.
- خط ساحلی (*beach line*) تابعی است که از پایین ترین نقطه همه سهمی ها عبور می کند.



خواص beach line

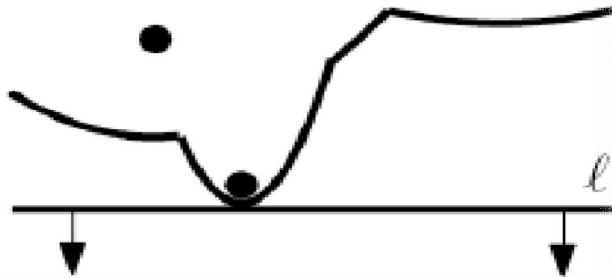
- **beach line** نسبت به محور Xها یکنواست. به این معنی که برای هر خط عمودی آن را دقیقاً در یک نقطه قطع می کند.
- یک سهمی ممکن است در دو قسمت از **beach line** ظاهر شود
- محل تقاطع سهمی ها روی یک ضلع دیاگرام ورونوی قرار دارد
- **break point** میان نمودارهای سهمی مختلف روی یالهای دیاگرام ورونوی قرار می گیرند
- نقاط شکست دقیقاً دیاگرام ورونوی را رسم می کنند درحالیکه **sweep line** از بالا به پایین حرکت می کند



تغییرات beach line

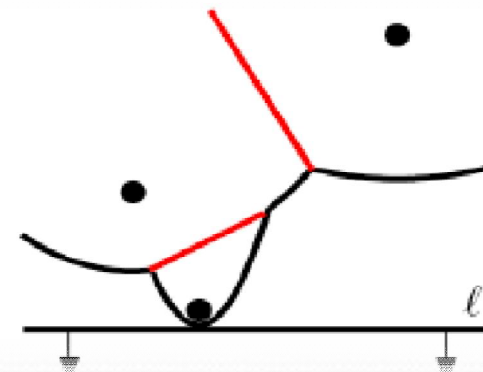
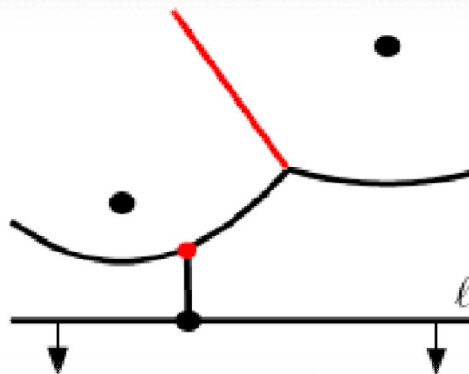
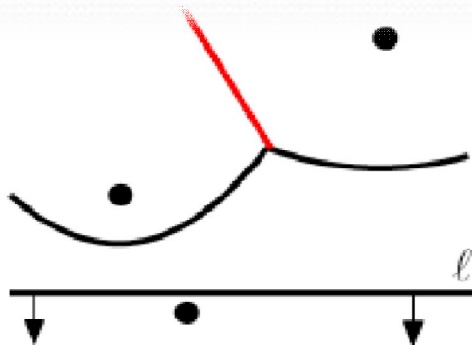
● **site event**: وقتی **sweep line** به یک سایت

می رسد یک پاره خط که با پایین رفتن l سهمی باز می شود



● **تغییرات دیاگرام ورونوی**:

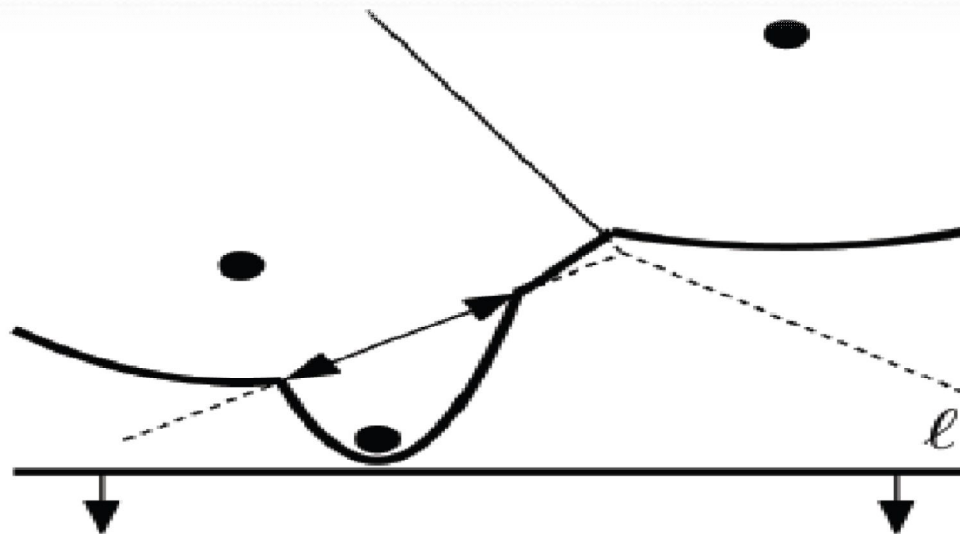
ابتدا یک نقطه و سپس یک پاره خط جدا از بقیه دیاگرام اضافه می شود که به تدریج به بقیه دیاگرام متصل می شود

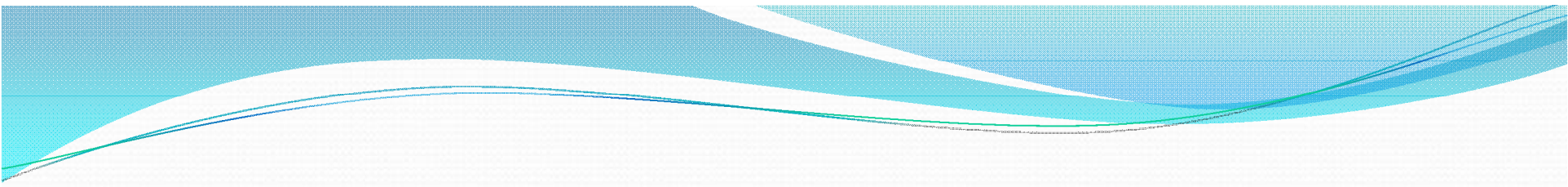


چه اتفاقی برای دیاگرام ورونوی در **site event** می افتد؟

در یک **site event** دو نقطه جدید ظاهر می شود که شروع به رسم کردن یالها می کند:

- دو نقطه شکست در ابتدا بر هم منطبقند
- سپس در خلاف جهت برای رسم همان یال حرکت می کنند





- **لم :** تنها راهی که کمان جدید می تواند در *beach line* ظاهر شود توسط یک *site event* است.

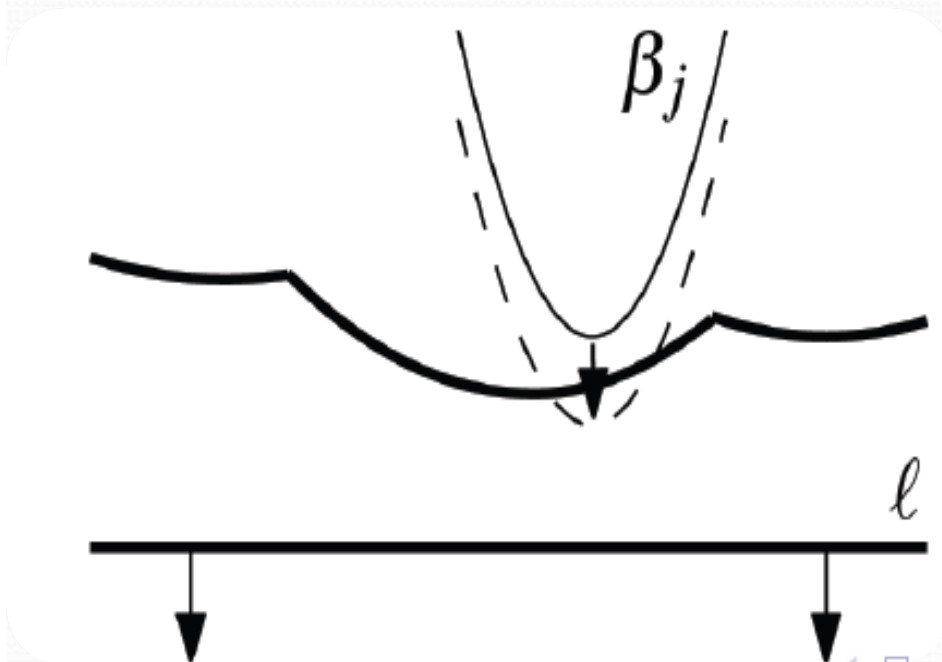
اثبات: فرض کنید سهمی β_j موجود توسط سایت p_j تعریف می شود و *beach line* را می شکند
دو حالت وجود دارد که این اتفاق می افتد:

- حالت اول این است که β_j در میان یک سهمی همچون β_i باشد

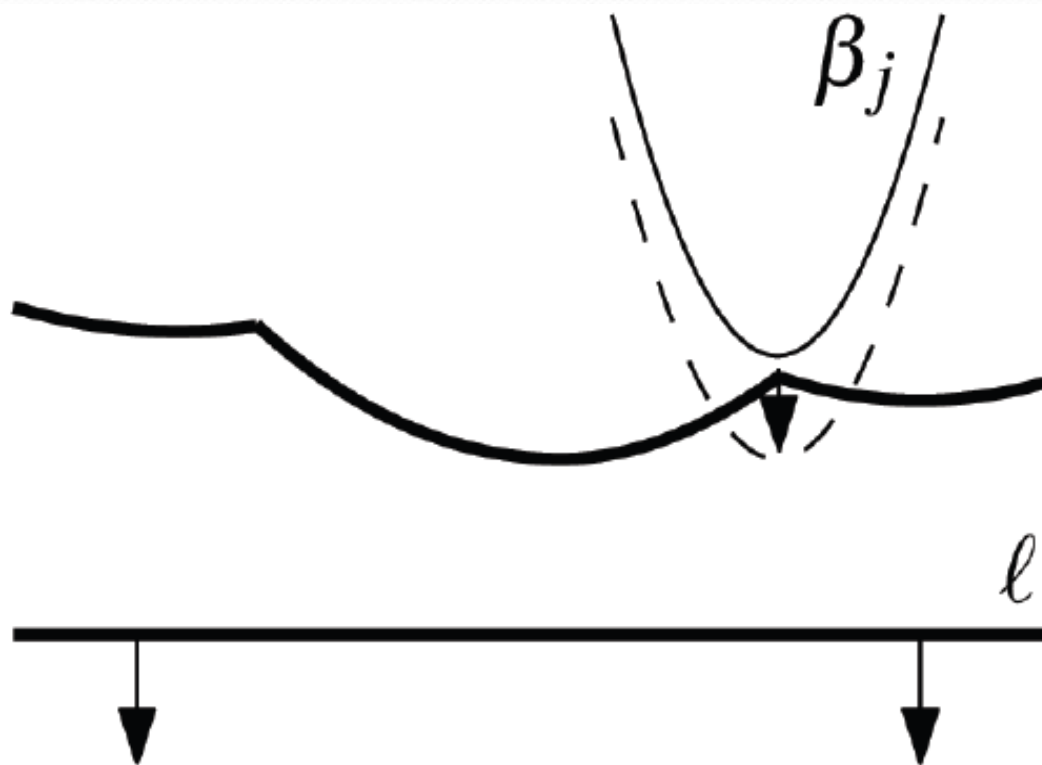
l_y : مختص y از *sweep line* را نشان می دهد

$$p_j := (p_{j,x}, p_{j,y})$$

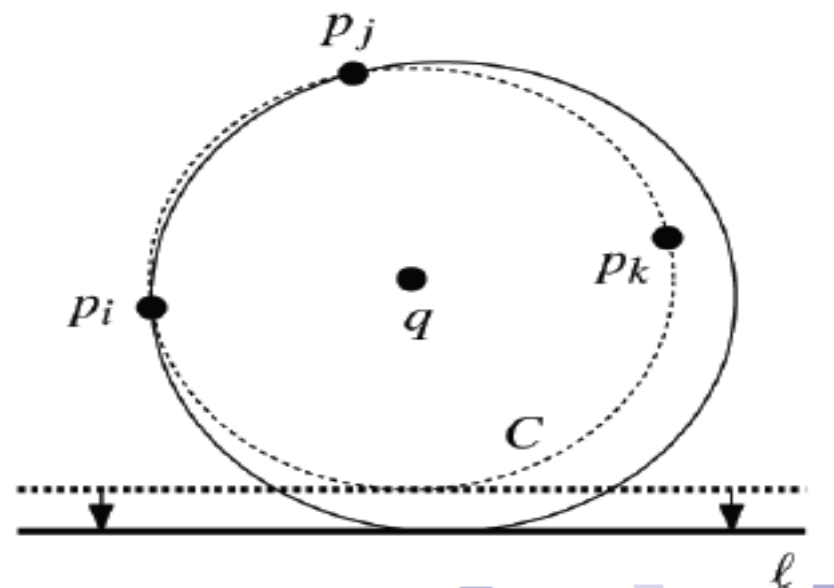
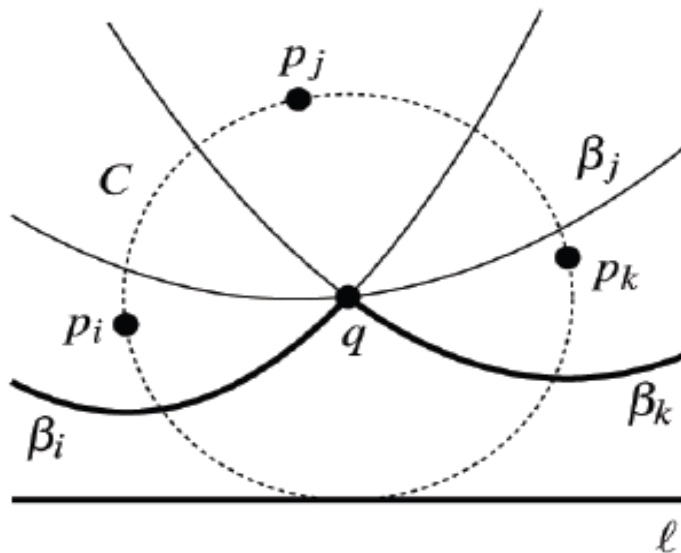
$$\beta_j := y = \frac{1}{2(p_{j,y} - l_y)}(x^2 - 2p_{j,x}x + p_{j,x}^2 + p_{j,y}^2 - l_y^2)$$



- حالت دوم این است که β_j در بین دو منحنی β_k و β_l ظاهر شود:



- q : نقطه برخورد از β_i و β_k است
- β_j حول نقطه q روی $beach\ line$ ظاهر می شود
- β_i در $beach\ line$ سمت چپ q
- β_k در $beach\ line$ سمت راست q



یک نتیجه مهم از لم

- *beach line* حداکثر شامل $2n - 1$ منحنی سهمی است. زیرا اضافه شدن هر سایت جدید یک سهمی جدید تولید می کند که حداکثر یکی از سهمی های قبل را به دو قسمت تقسیم می کند.

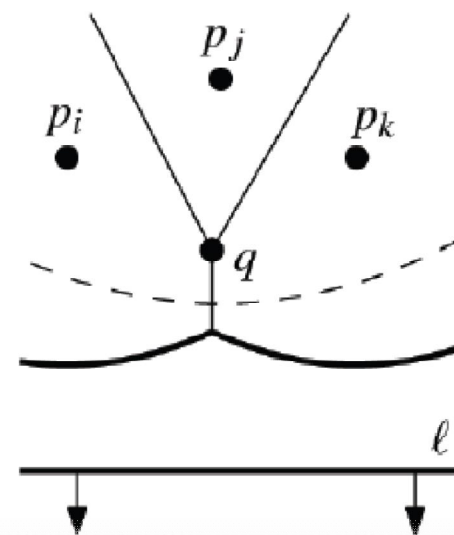
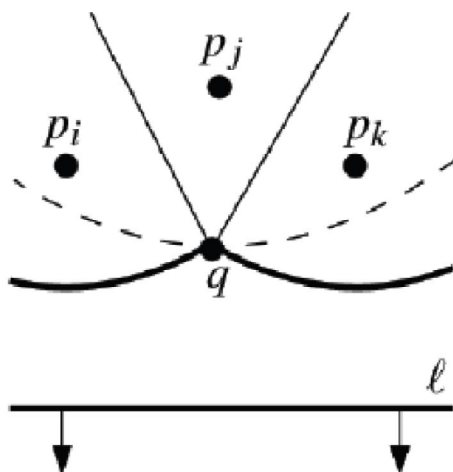
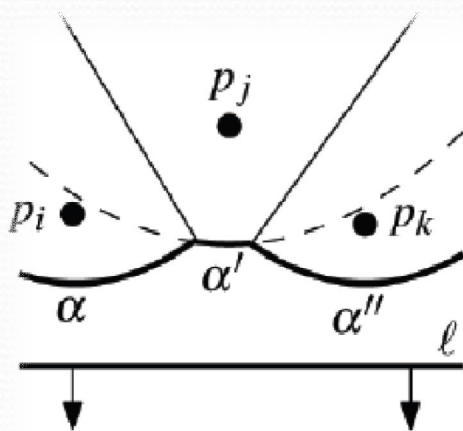
- دومین نوع از *event* در الگوریتم *plan sweep* این است که یک کمان از *beach line* به یک نقطه کاهش یابد

فرضیات:

α' : کمانی باشد که حذف می شود

α و α'' : دو کمان همسایه α' قبل از حذف شدن آن باشند

q : نقطه متساوی الفاصله از ℓ و هر یک از سه سایت است

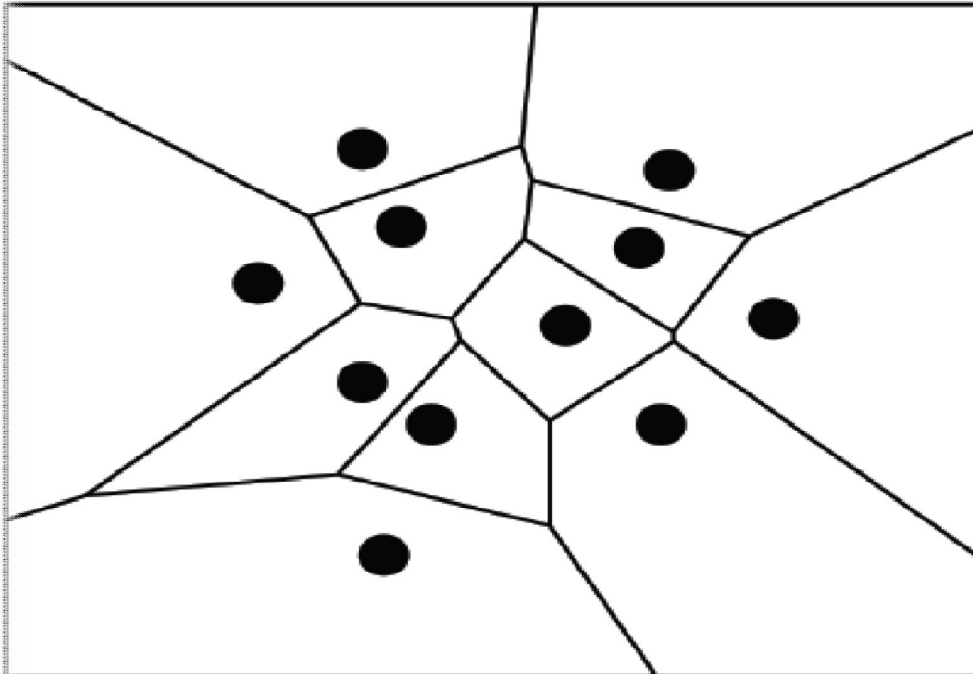


نکته:

- در زمانی که یک سهمی از *beach line* حذف شود (*circle event*) یک راس در دیاگرام ورونوی تولید می شود
- در هر *site event* یک ضلع جدید ایجاد شده رشد می کند تا به بقیه دیاگرام متصل شود
- در هر *circle event* یک راس تولید می شود

ساختمان داده مورد نیاز

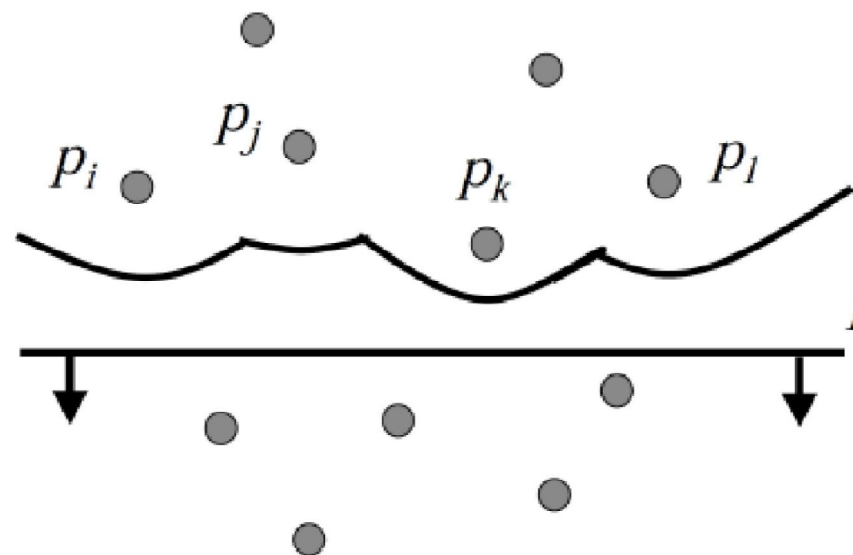
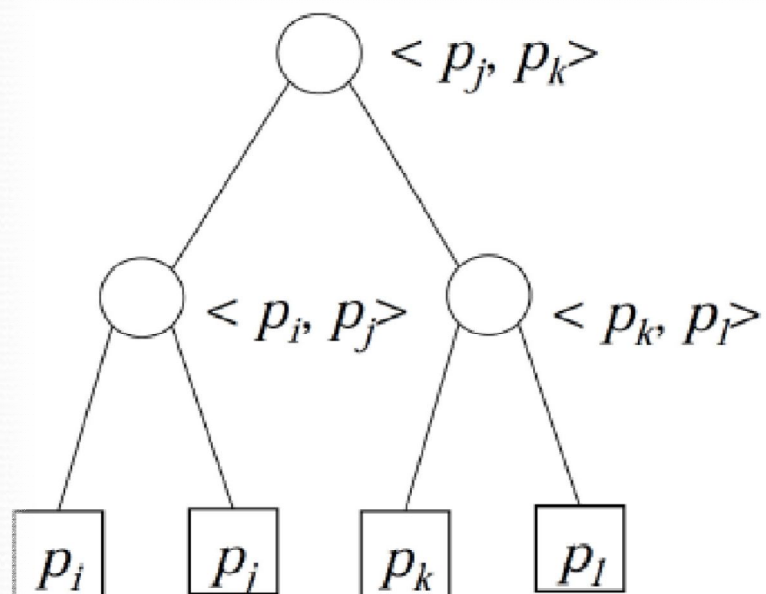
- یک ساختمان داده که قسمتی از دیاگرام ورونوی که تا کنون محاسبه شده را ذخیره کند
- اگر دیاگرام ورونوی را بایک مستطیل کراندار کنیم، یک زیر تقسیم به دست می آید که با DCEL ذخیره می شود



ساختمان داده مورد نیاز

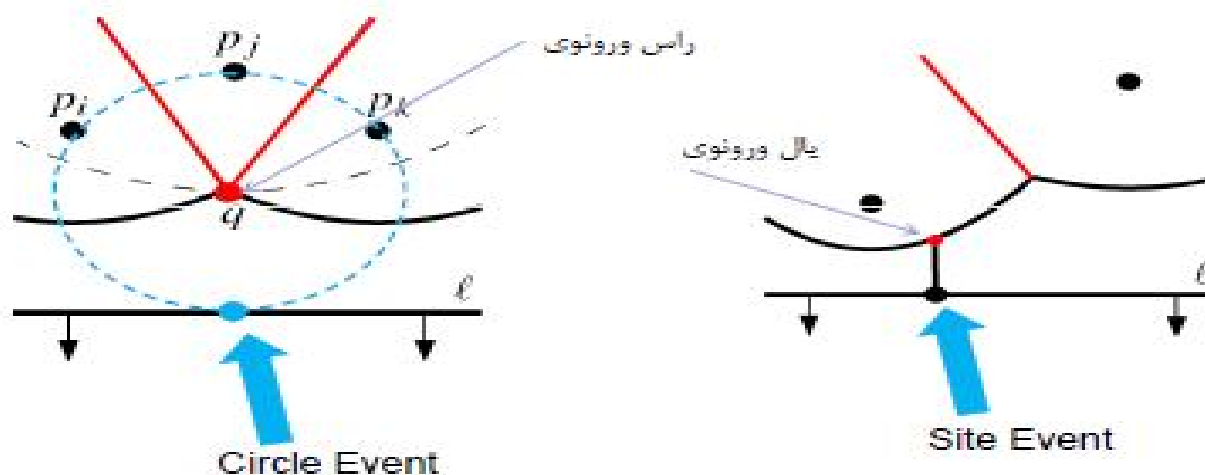
- برای الگوریتم *sweep line* :
- یک درخت جستجوی دودویی برای حفظ وضعیت *sweep line* (که این جا *beach line*)
- یک صف از پیشامدها

● *beach line* با یک درخت جستجوی دودویی متوازن نشان داده می شود



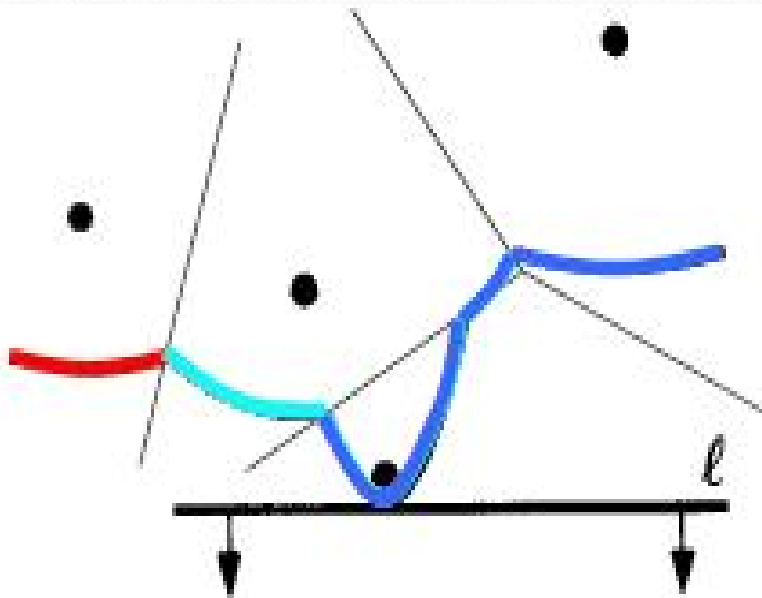
صف پیشامدها

- پیشامدهایی که تا کنون مشخص شده اند در صف ذخیره می شوند
- پیشامدها در صف بر اساس مختصات y آن ها ذخیره می شوند
- در هر **site event** خود سایت ذخیره می شود
- در هر **circle event** پایین ترین نقطه دایره به اضافه یک اشاره گر به برگگی از درخت جستجوی دودویی که کمانی که در این پیشامد حذف می شود را نشان می دهد، ذخیره می شود



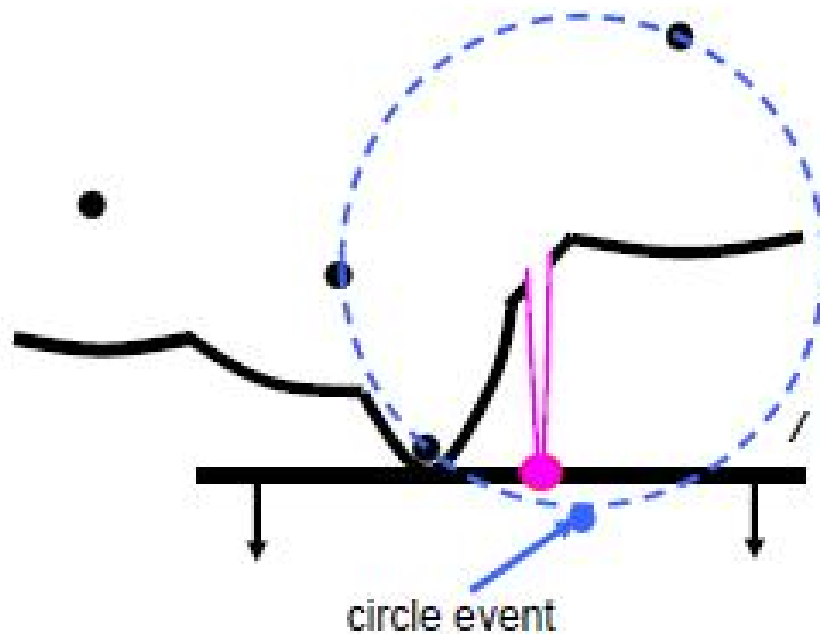
تعین circle event

- تعیین circle event ها کمی پیچیده تر است
- در هر پیشامد همه سه تایی های جدید کمان های متوالی beach line بررسی می شوند
- در یک سه تایی کمان جدید سمت راست در یکی وسط و در دیگری در سمت چپ قرار دارد
- اگر نقاط تلاقی سهمی ها به هم نزدیک شوند ممکن است یک circle event به وجود آید
- پس سه تایی دوم circle event تولید نمی کند



تعین circle event

- اگر نقاط تلاقی سهمی ها به هم نزدیک شوند ممکن است یک circle event نباشد



Algorithm VORONOIDIAGRAM(P)

Input. A set $P := \{p_1, \dots, p_n\}$ of point sites in the plane.

Output. The Voronoi diagram $\text{Vor}(P)$ given inside a bounding box in a doubly-connected edge list $\mathcal{D}$.

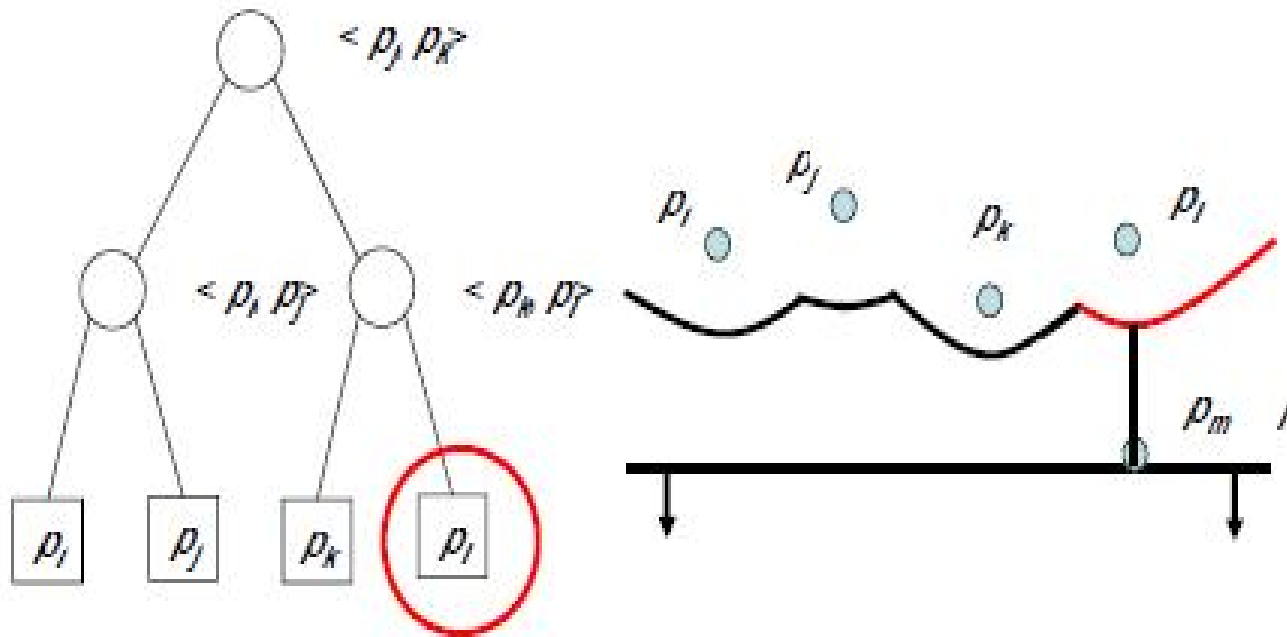
1. Initialize the event queue $\mathcal{Q}$ with all site events, initialize an empty status structure $\mathcal{T}$ and an empty doubly-connected edge list $\mathcal{D}$.
2. **while** $\mathcal{Q}$ is not empty
3. **do** Remove the event with largest y-coordinate from $\mathcal{Q}$.
4. **if** the event is a site event, occurring at site p_i
5. **then** HANDLESITEEVENT(p_i)
6. **else** HANDLECIRCLEEVENT(γ), where γ is the leaf of $\mathcal{T}$ representing the arc that will disappear
7. The internal nodes still present in $\mathcal{T}$ correspond to the half-infinite edges of the Voronoi diagram. Compute a bounding box that contains all vertices of the Voronoi diagram in its interior, and attach the half-infinite edges to the bounding box by updating the doubly-connected edge list appropriately.
8. Traverse the half-edges of the doubly-connected edge list to add the cell records and the pointers to and from them.

HANDLESITEEVENT(p_i)

1. If $\mathcal{T}$ is empty, insert p_i into it (so that $\mathcal{T}$ consists of a single leaf storing p_i) and return. Otherwise, continue with steps 2–5.
2. Search in $\mathcal{T}$ for the arc α vertically above p_i . If the leaf representing α has a pointer to a circle event in $\mathcal{Q}$, then this circle event is a false alarm and it must be deleted from $\mathcal{Q}$.
3. Replace the leaf of $\mathcal{T}$ that represents α with a subtree having three leaves. The middle leaf stores the new site p_i and the other two leaves store the site p_j that was originally stored with α . Store the tuples $\langle p_j, p_i \rangle$ and $\langle p_i, p_j \rangle$ representing the new breakpoints at the two new internal nodes. Perform rebalancing operations on $\mathcal{T}$ if necessary.
4. Create new half-edge records in the Voronoi diagram structure for the edge separating $\mathcal{V}(p_i)$ and $\mathcal{V}(p_j)$, which will be traced out by the two new breakpoints.
5. Check the triple of consecutive arcs where the new arc for p_i is the left arc to see if the breakpoints converge. If so, insert the circle event into $\mathcal{Q}$ and add pointers between the node in $\mathcal{T}$ and the node in $\mathcal{Q}$. Do the same for the triple where the new arc is the right arc.

اجرای الگوریتم site event

با استفاده از مختصات x نقطه پیشامد جدید در درخت جستجوی دودویی
کمانی که بالای نقطه پیشامد است را پیدا می کنیم



- درخت جستجوی دودویی (beach line) را به روز می کنیم

