Complexity estimates for trapezoidal map

(1) Randomized running time for searching

(2) Randomized size of search structure

(3) Randomized time for the construction of the search structure

(1) Searching. The time depends on the length of passes in the search structure. We have the search for n segments. X_i is random variable for the number of new nodes which are created in the i-th step when we go from the root to the leaf.

$$0 \leq X_i \leq 3$$

Randomized running time is

$$E\left(\sum_{i=1}^{n} X_i\right) = \sum_{i=1}^{n} E X_i$$

$$E(X_i) = 0 \cdot p(X_i = 0) + 1 \cdot p(X_i = 1) + 2 \cdot p(X_i = 2) + 3 \cdot p(X_i = 3)$$

$$E(X_i) \leq 3 \cdot \underbrace{p(X_i \neq 0)}$$

$$p(X_i \neq 0) = p\Big(\text{the given point lies in the trapezoid in } T_i \text{ which is determined by } s_i\Big\}$$

$$= p\Big(\text{given point lies in } \Delta \text{ and } \mathrm{top}(\Delta) = s_i \text{ or } \mathrm{bottom}(\Delta) = s_i \text{ or } \mathrm{rightp}(\Delta) = q_i \text{ or } \mathrm{leftp}(\Delta) = p_i\Big) \leq \frac{4}{i}$$

$$\sum_{i=1}^{n} E(X_i) \leq \sum_{i=1}^{n} 3 \cdot \frac{4}{i} = 12\left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}\right) \leq$$

$$\leq 12(1 + \log n) = O(\log n)$$

(2) Size of the search structure

Number of leaves + number of internal nodes

$$3n+1 \quad + \sum_{i=1}^{n} \text{number of internal nodes arising in the } i\text{-th step}$$

$$= 3n+1 \quad + \sum_{i=1}^{n} (k_i - 1)$$

where k_i is the number of new trapezoids arising in the i-th step.

Randomised size $\leq O(n) + \sum_{i=1}^{n} E(k_i)$

(4)

$\Delta \in T(S_i)$ a trapezoid

$s \in S_i$ a segment

$$\lambda(\Delta, s) = \begin{cases} 1 & \text{if } \Delta \text{ arises by adding } s \\ 0 & \text{otherwise} \end{cases}$$

$$\sum_{s \in S_i} \lambda(\Delta, s) \leq 4$$

$$m_i = \sum_{\Delta \in T(S_i)} \lambda(\Delta, s_i)$$

$$\sum_{s \in S_i} \sum_{\Delta \in T(S_i)} \lambda(\Delta, s) \leq 4 \cdot \text{number of trapezoids in } T(S_i)$$

$$\leq 4(3i+1) = O(i)$$

$$E(m_i) = \frac{1}{i} \sum_{s \in S_i} \underbrace{\left(\sum_{\Delta \in T(S_i)} \lambda(\Delta, s) \right)}_{m_i(s)} = \frac{O(i)}{i} = O(1)$$

Randomized size

$$\leq O(n) + \sum_{i=1}^{n} E(n_i) \leq O(n) + n\, O(1) = O(n)$$

3 Construction of the search structure

Time for creating $T(S_i)$ and $\mathcal{D}(S_i)$ from $T(S_{i-1})$ and $\mathcal{D}(S_{i-1})$ is given by $\underbrace{\text{searching } \Delta_0 \text{ which is crossed by } s_i}_{O(\log i)}$ and by $E(n_i)$.

Randomized time for the construction

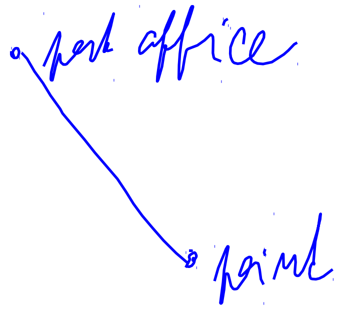
$$\leq \sum_{i=1}^{n} \left(E(n_i) + O(\log i) \right) \leq n \cdot O(1) + n\, O(\log n) = O(n \log n)$$

Voronoi diagrams

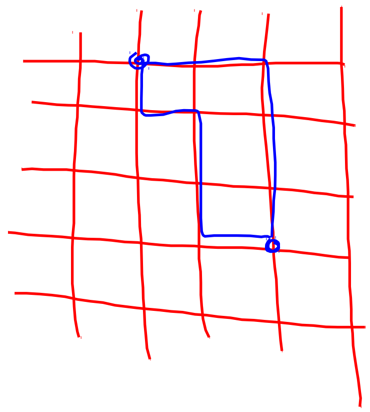
Post office problem – a city with many post offices; we want to divide the territory of the city into areas around post offices in such a way that every point from any area has the closest post office in this area.

ℓ_2

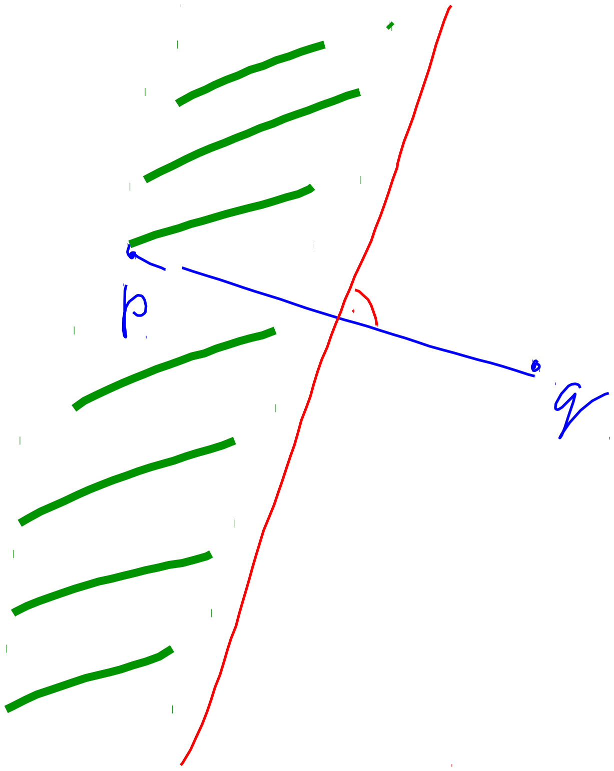
$$d(p,q) = \sqrt{(p_x-q_x)^2+(p_y-q_y)^2}$$



ℓ_1 metric



$$d(p,q) = |p_x-q_x| + |p_y-q_y|$$



7

$$P = \{p_1, p_2, \dots, p_n\}$$

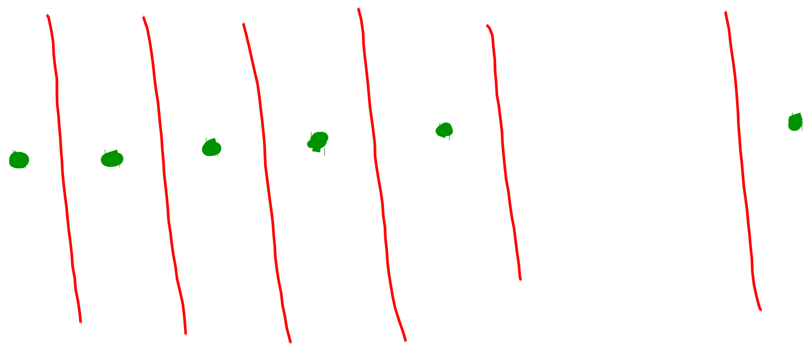
$$h(p_i, p_j) = \{q \in \mathbb{R}^2,\ d(q, p_i) \le d(q, p_j)\}$$

$$i \neq j$$

$$V(p_i) = \text{Voronoi cell}$$

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

Good algorithms compute Voronoi diagram in the time $O(n \log n)$. Better time is impossible.

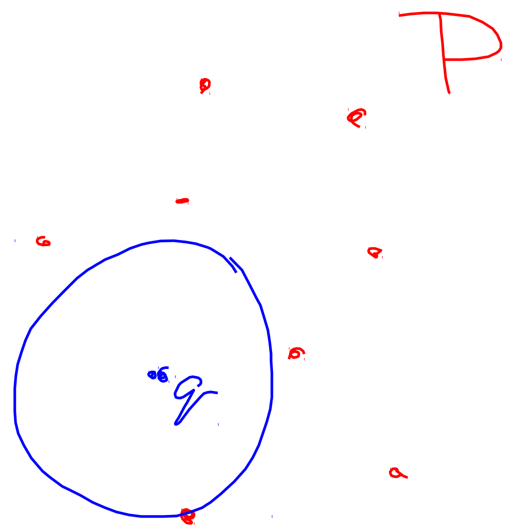


More methods how to compute Voronoi diagram

— sweep line method

$P = \{p_1, \dots, p_n\}$

$q \in \mathbb{R}^2$ $\quad C_P(q) = \{r \in \mathbb{R}^2,\ dist(q, r) \le dist(q, P)\}$



$C_P(q)$ always contains a point from P

If it contains two points, q lies on the edge of the Voronoi diagram. If it contains 3 points, q is a vertex of Voronoi diagram.

These conditions are not only necessary but also sufficient

— see picture 9.3

<u>Thm</u>: The number of vertices of V-diagram for n points is at most $2n-5$, the number of edges is at most $3n-6$.

$(n \geq 3)$

Planar graph with $m+1$ points
k edges
n faces

$$\boxed{m+1-k+n=2}$$

In every vertex at least 3 edges meet,

$2k$ = the sum of degrees of all vertices

$$2k \geq 3(m+1) \qquad k = m-1+n$$

$$2(m-1+n) \geq 3m+3$$

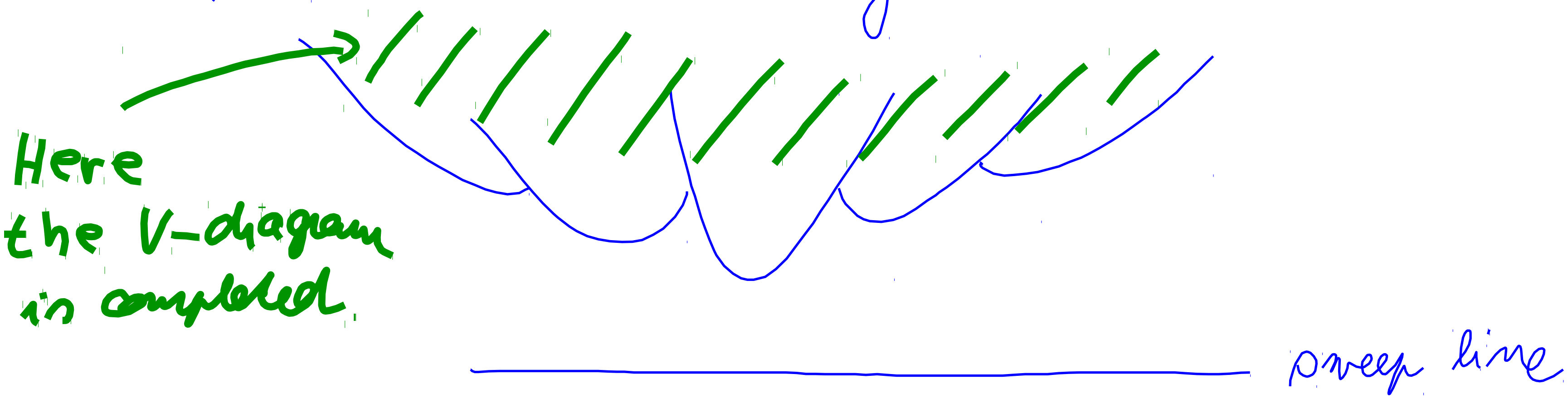
$$2n-2-3 \geq m$$

similarly $m = k+1-n$ gives $k \leq 3n-6$.

Sweep line method

Usually we want to find or create something and this is created (computed) for objects over the sweep line.

Here the V-diagram is created over so called beach line which is bounded by arcs of parabolas.

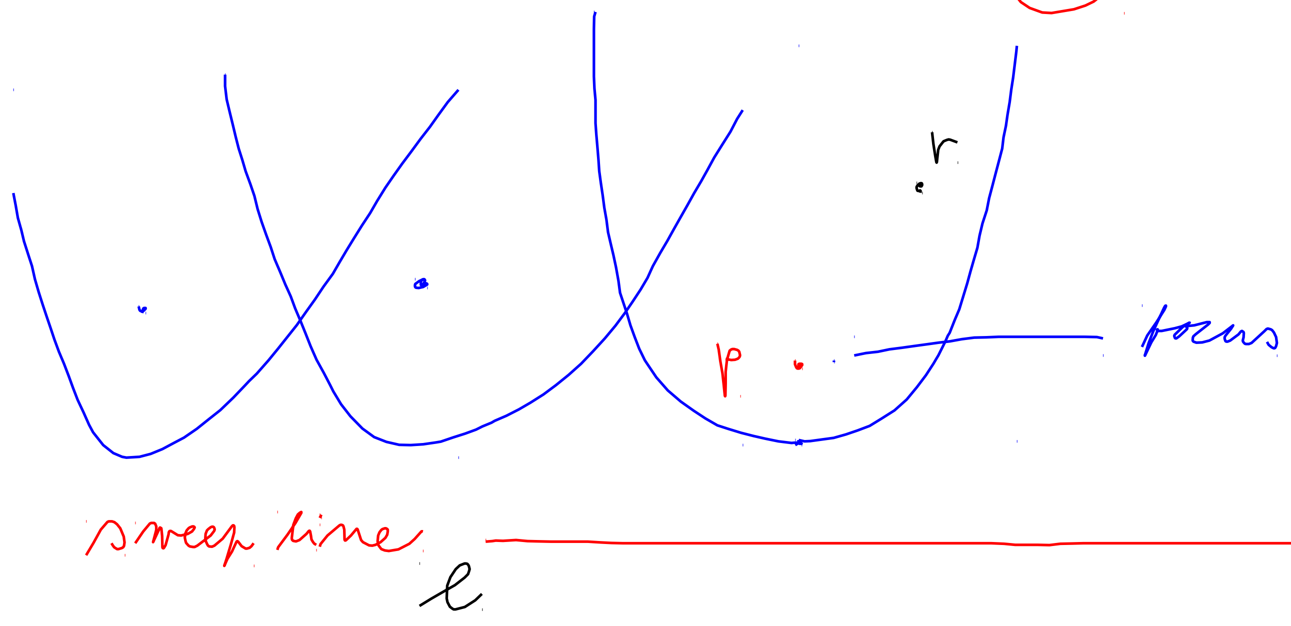


Parabola is the location of points which have the same distance from given point and a given line.

10

$p \in P$

$q \in P$



$\mathrm{dist}(r, p) \le \mathrm{dist}(r, \ell) \le \mathrm{dist}(r, q)$

q has no influence on the Voronoi diagram over the parabola

$\alpha(p, \ell)$ ... parabola or an arc of parabola with focus in p and directrix ℓ

$\alpha^{+}(p, \ell)$ ... parabola together with the points over the parabola

Voronoi diagram in this algorithm is constructed for the union $\bigcup_{p \in \ell^+} \alpha^+(p, \ell)$ p lies over the sweep line ℓ^+

Beach line is the boundary of the previous area, formed by arcs of parabolas.

Breakpoints are points where two arcs meet.

Notation $\langle p_1 : p_2 \rangle$

Balanced binary tree for given position of sweep line is given by the order of arcs in the beach line from the left to the right.

Queue .... points = events are in lexicographic order

$p > q \quad p_y > q_y$ or $p_y = q_y$ and $p_x < q_x$

At the beginning all the points from P are in the queue.

During algorithm some other points are added to the queue.

Two kinds of events

Site events .... in these poits a new arc arises in the beach line

Lemma: The only way how a new arc appear in the beach line is when the sweep line crossed a point from the set P. Site events

Maximal number of arcs in the beach line is

$$1 + \underbrace{2 + \dots + 2}_{n-1} \leq 2n - 1$$

<u>Lemma</u> Let α, β, γ be 3 arcs of the beach line with focuses p_1, p, p_3. Let C be a circle circumscribed the [illegible] $_2 p_3$, s its center and q the point with the smallest y-coordinate. Then the only way how β can dissapear from the beach line is when the sweep line crosses the point q.

q is called <u>circle event</u>