Computational Geometry (CSE 721)

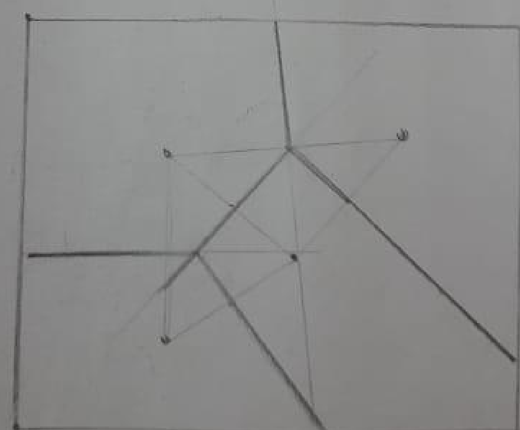
(Group - 12)

Name: Sahil Kumar, Manish Kumar Singh, Yug Soni

Roll No: : 20CS8038, 20CS8031, 20CS8087.

Describe an algorithm that determine in linear time, given a Voronoi diagram with n region, the set of 'n' points that define it. (You should be able to describe your algorithm purely geometrically, without using any of ugly algebra that would show up in actual implementation). What does your algorithm do if we give it a planar map that is not a Voronoi diagram?

- A voronoi diagram is a geometric structure that divides a plane into region based on the proximity of a specified set of points called sites.



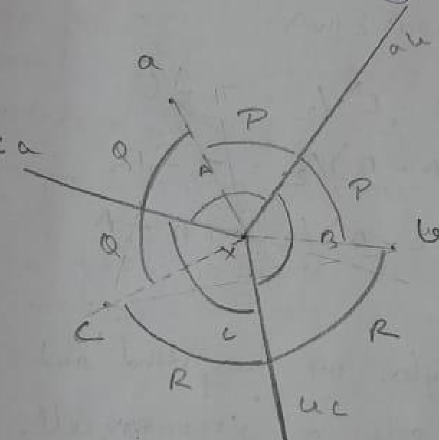
- There are following features of a voronoi diagram which we will be using to solve above problem.

P1 → For $n \geq 3$, the number of vertices in the Voronoi diagram of a set of 'n' point site in the plane is at most $2n-5$ and the number of edges is at most $3n-6$.

P2 → The edges in a voroni diagram is a perpendicular bisector of edge joing sites on the either sides of the edge.

Intuition:

For a single intersection of the Voronoi Diagram, you will generally have 3 edges and 3 sectors. between edges; Call the sector (and their angle) 'A', 'B', 'C'. Also call edge b/w sector 'A' and 'B' the edge 'ab', and likewise for edge 'bc' and 'ca'.



From the property (P2) we can say that 'ab' edge is a ⊥r bisector of line joing pt. 'a' & 'b'. So, the ∠ b/w $a\,x\,\vec{ab}$ is same as ∠ b/w $b\,x\,\overline{ab}$. Let this angle be 'P'.

Similary for site 'a', 'c' and edge 'ca' the angle ~~b/w~~. $\angle a\,x\,\overline{ca}$ and $\angle c\,x\,\overline{ca}$ is 'Q'.

and for site 'c', 'b' and edge 'bc' the angle b/w $\angle c\,x\,\overline{bc}$ and $\angle b\,x\,\overline{bc}$ is 'R'

We can say that

$$A = P + Q \quad \text{— (i)}$$

$$B = R + P \quad \text{— (ii)}$$

$$C = Q + R \quad \text{— (iii)}$$

$$A + B + C = 2\pi \quad \text{— (iv)}$$

From (i), (ii), (iii) & (iv)

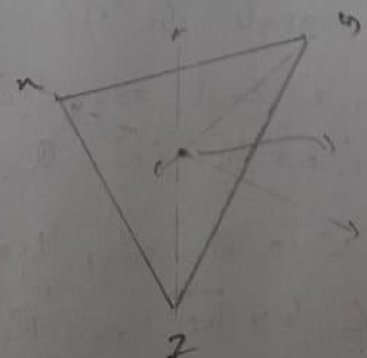
$$P = (A+B-C)/2 = \pi - C$$

$$Q = (C+A-B)/2 = \pi - B$$

$$R = (B+C-A)/2 = \pi - A$$

By using these angles we can find out a ray which passes through site a voronoi cell.

And by ~~some~~ using the intersection of all the vertices of a given voronoi cell, we will the site of that particular cell.



~~vertices~~ site of this cell

...gorithm:

INPUT: an adjacent list that gives all the connecting vertices from a given vertice through edges.

OUTPUT: a set of point P which represents the site of voronoi diagram.

1) Initialize an event queue 'Q' that store all event point and unordered map ~~that~~ 'MP' that store the ray passing through vertice and site, where key is vertice and value is ray.

all given vertice will be inserted into Q.
We will also initialize 'R' to store result.

2) While Q is no empty:

3) if next event is a vertice:

4) Handle vertice

5) else.

Handle ray.

6) Return ~~rest.~~ 'R'.

The procedure to handle procedure defin follows:

Handle vertice (V_i) :

1) By following the intuition we will create a ray passing through vertice V_i ~~and~~ in (angle will be in anti clockwise order) for all the neighbour vertices

(we will use some data structure to track wether the ~~following~~ current combination as been computed before, so that all the edges will be consider for only one time for the vertex V_i)

2) We will add all these rays in MP and in Q.

3) Delete V_i from Q.

Handle ray (R_i)

1) We ~~will~~ had stored our ray R_i and its correspon-ding vertex (V) . So using adjacent list we will find the intersection with all the neighbour vertices ray.

2) Report all ~~those~~ these ray in R

3) remove R_i from Q.

Time - complexity analysis:

- From property (P1) we can say that number of edges and no. of sites are linearly co-related. (linear time)
- Using ammortized analysis we can say that for every ~~there~~ vertex can be a part of constant number of cells.
- Handle vertice (V_i) is basically creating ~~vertice~~ ray from a V_i vertex. (constant time)
- Handle ray (R_i) construct side by using ~~tasker~~ neighbour-hood rays (This can be optimized by considering and angle made by neighbour vertices) (constant time)

So. over all time complexity is $O(n) * k * m$

$$\approx O(n)$$

Working of this algo for non-voronoi diagram:

. Since, ~~the~~ if the given diagram isn't voronoi so our induction will be wrong and then the result will consiste of sides belonging to the same site.