Recall

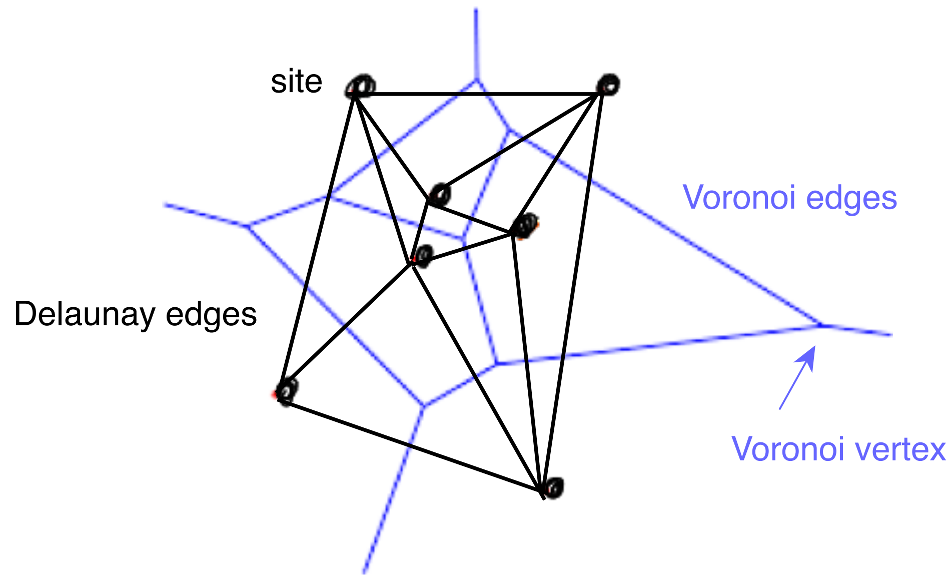
Voronoi diagram

Given points $P = \{p_1, . . . , p_n\}$ in the plane, the ***Voronoi region*** of p_i is

$$V(p_i) = \{x \in \mathbb{R}^2 : d(x, p_i) \leq d(x, p_j) \forall j \neq i\}$$

p_i is called a ***site***.

The ***Voronoi diagram*** $\mathcal{V}(P)$
consists of all the Voronoi regions

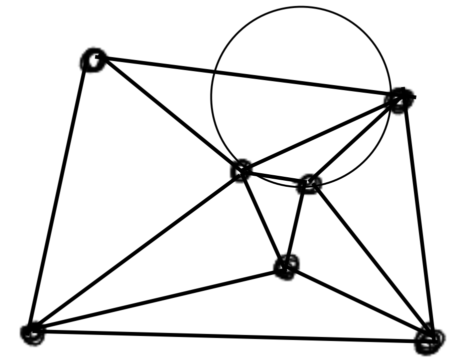


Given points $P = \{p_1, . . . , p_n\}$ in the plane, the ***Delaunay triangulation*** $\mathcal{D}(P)$ is a graph with vertices $p_1, . . . , p_n$ and edge (p_i, p_j) iff $V(p_i)$ and $V(p_j)$ share an edge.

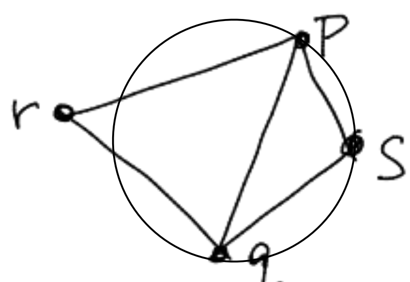
$\mathcal{D}(P)$ is the ***planar dual*** of $\mathcal{V}(P)$

Recall Delaunay triangulation
and empty circle property: (p,q) is an edge of the Delaunay triangulation iff there is an empty circle through p and q.

An algorithmically more useful characterization:



Lemma. A triangulation is Delaunay iff every edge e=(p,q) is legal.



(p,q) is legal

Definition. edge e=(p,q) is ***legal*** if either:
- e is on the convex hull or
- e is interior with triangles pqr and pqs, and r is not in Circle(pqs)

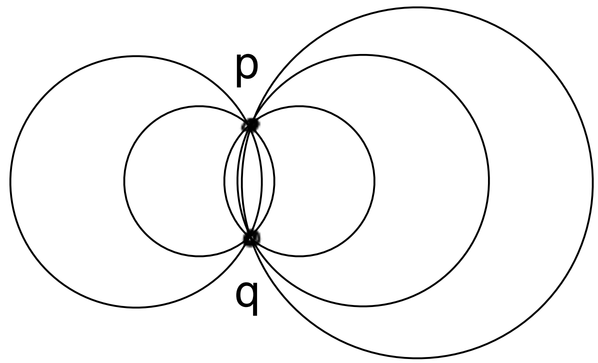
Note: r in Circle(p,q,s) iff s in Circle(p,q,r)

Note that this is a condition about ALL edges, not a single edge:

edge e is Delaunay (∃ an empty circle through its endpoints) $\Rightarrow$ e is legal
$\not\Leftarrow$

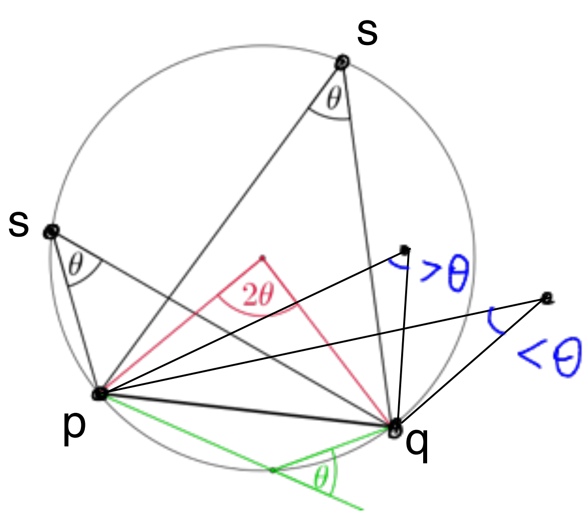
Ingredients to prove the lemma.

There is one degree of freedom for circles through p and q.

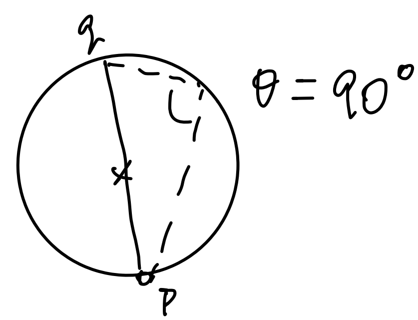


Thales Theorem. For pq a chord of a circle, angle psq is constant for s on an arc of the circle. For s inside, the angle is bigger. For s outside the angle is smaller.

Actually, we only need this later on (not for this lemma)



W https://en.wikipedia.org/wiki/Inscribed_angle#Theorem



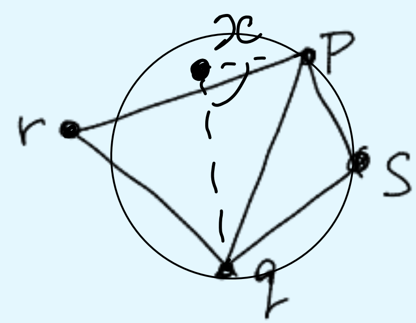
Actually, Thales considered pq to be a diameter. The generalization is in Euclid.

Lemma. A triangulation is Delaunay iff every edge e=(p,q) is legal.

Proof. contrapositive

triang. is NOT Delaunay iff $\exists$ an illegal edge.

$\Leftarrow$ an illegal edge p,q has no empty circle
$\therefore$ not Delaunay.



(p,q) is legal

$\Rightarrow$ then $\exists$ triangle pqs and site x in Circle(pqs)

Choose pqs and x to maximize $\angle pxq$

Let r be vertex of triangle on other side from pqs.

If r in Circle(pqs) then (p,q) is illegal.

Note x not in $\triangle pqr$. Suppose x outside edge rp

Consider pqr and point x.

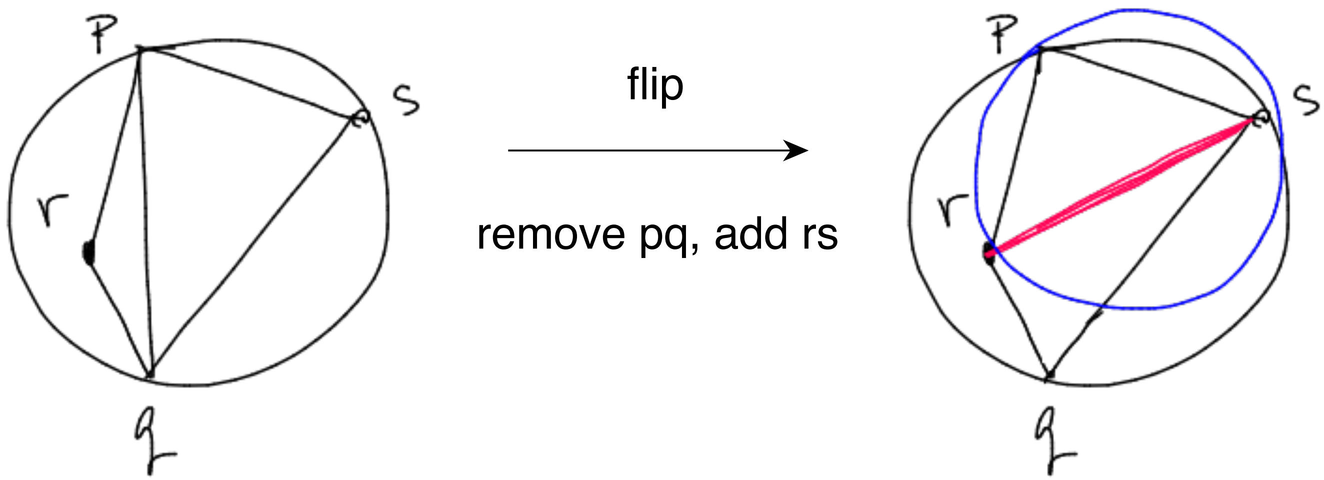
$x \in$ Circle(pqr)

and $\angle pxr > \angle pxq$. Contradiction to choice of pqs and x.

? Where did we use Thales. — we didn't! though this pf. is straight from [CGAA]. We will use Thales later.

What to do with an illegal edge (p,q)

Edge Flip



Claim: (r,s) is a legal edge.

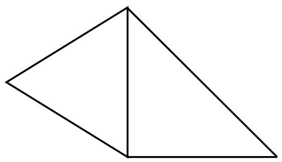
change Circle (pqs) to Circle (prs)
it shrinks and q leaves the circle.

Flipping illegal edges makes global improvements in a triangulation:
the **Angle Vector**.

For any triangulation T of a set of points, the ***angle vector*** A(T) is the list of angles of the triangles sorted min to max.

example

T



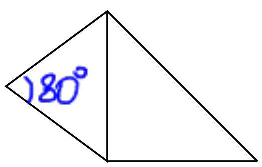
A(T) = (45, 45, 60, 60, 60, 90)

The angle vector always has length 3t where t is the number of triangles.

We compare two angle vectors ***lexicographically*** (dictionary ordering)

example

T'



A(T') = (45, 45, 50, 50, 80, 90)

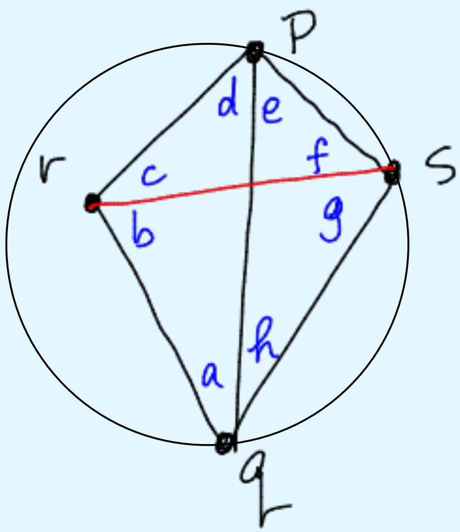
$\parallel \quad \parallel \quad \wedge$

A(T) = (45, 45, 60, 60, 60, 90)

so A(T) > A(T')

Lemma. Flipping an illegal edge increases the angle vector lexicographically.

Proof.



flip (p, q) to (r, s).

before flip: $a, d, h, e, b+c, f+g$ (in some order)

after flip: $f, g, c, b, d+e, a+h$ " "

$c > h$ using Thales on chord ps

$b > e$ . . . . . . qs

$f > a$, $g > d$

min angle before flip: one of a, d, h, e
(because $b+c > e$, $f+g > a$)

claim: after flip every angle is > min before flip.

$f > a$ $g > d$ $c > h$ $b > e$

$d+e > d$ $a+h > a$

Note: other angle (outside $psqr$) don't change

$\therefore$ new min > old min.

Thus, flipping illegal edges **always** gets you to the Delaunay triangulation,
and the Delaunay triangulation has the lexicographically maximum angle vector.

Consequences:

Theorem. The Delaunay triangulation maximizes the minimum angle.

Algorithm to find the Delaunay triangulation: find ANY triangulation and then flip illegal edges until there are none left.

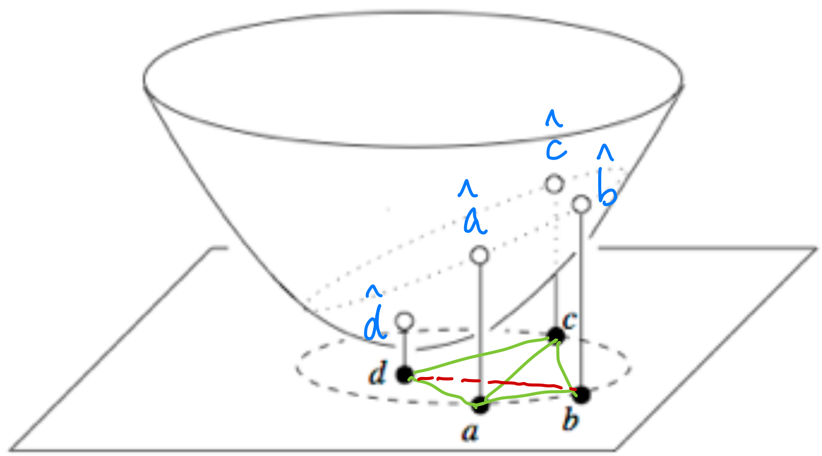
How many flips does this take?

We will prove that no edge reappears while flipping illegal edges.

Thus # flips $\leq \binom{n}{2}$ = # possible edges

Lemma. No edge reappears when we flip illegal edges.

Proof. Recall connection between Delaunay triangulation and convex hull



Flip illegal edge (a,c) to (d,b).

d inside Circle(abc) $\Longrightarrow$ $\hat{d}$ below plane($\hat{a}\hat{b}\hat{c}$)

$\hat{a}$ $\hat{b}$ $\hat{c}$ $\hat{d}$ form a tetrahedron in $\mathbb{R}^3$

Before flip have top 2 triangles $\hat{d}\hat{a}\hat{c}$ and $\hat{a}\hat{c}\hat{b}$

After flip have bottom 2 triangles $\hat{d}\hat{a}\hat{b}$ and $\hat{d}\hat{b}\hat{c}$

Segment $\hat{a}\hat{c}$ disappears forever above CH of raised points.

[Randomized] Incremental Delaunay triangulation algorithm.

Add points one by one, maintaining the Delaunay triangulation.
To add a new point p:

Find the current triangle ABC containing p.

Join p to A, B, C

Then flip illegal edges until there are none left.

Issues and details:

1. what if p is outside the current convex hull?

2. how to limit testing for illegal edges

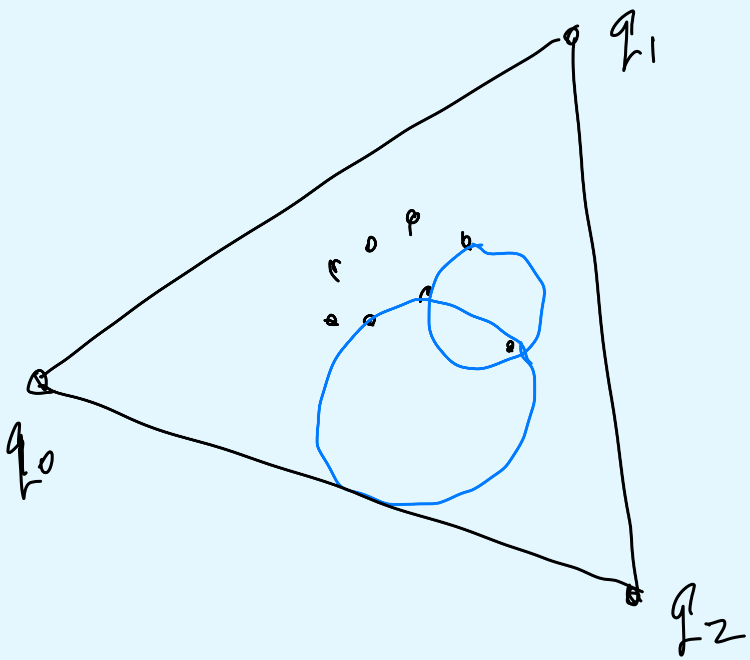
3. how to find the triangle containing p

[Randomized] Incremental Delaunay triangulation algorithm.

Issues and details.

1. what if p is outside the current convex hull?

Initialize by adding a very large triangle $q_0\, q_1\, q_2$ outside all points

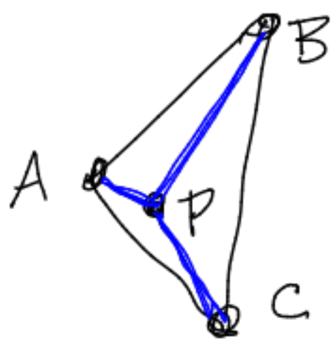


Condition:
every circle through 3 original points does not contain any q_i.

[Randomized] Incremental Delaunay triangulation algorithm.

Issues and details.

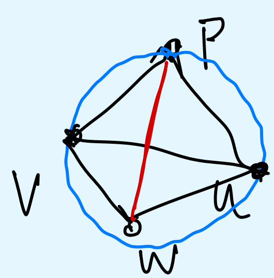
2. how to limit testing for illegal edges after adding point p



Call Test(A,B), Test(B,C), Test(C,A)

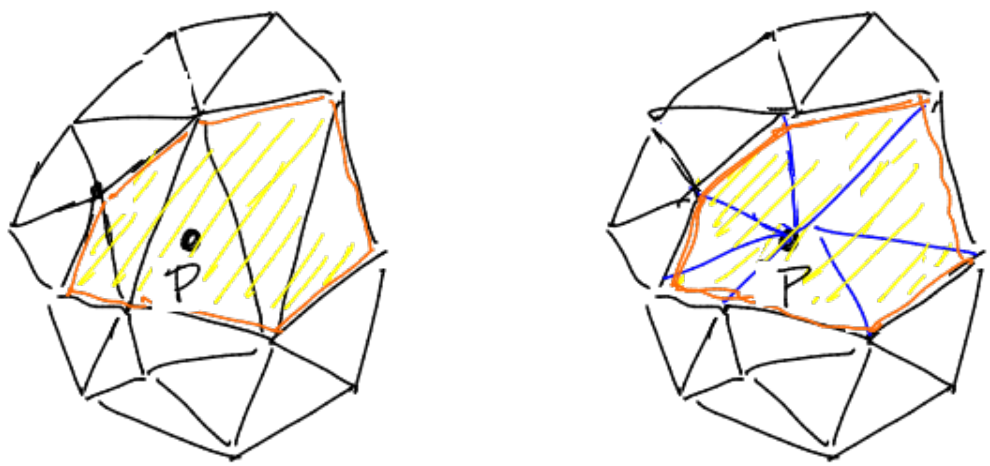
where Test(U,V) is a recursive routine
to fix edge UV in triangle UVp

Test(U,V) — if UV is illegal i.e. W is inside Circle(UVp)
where W is apex of triangle
on UV



- flip UV to pW
- test(U,W), Test(W,V)

Changes produced by this Test update:

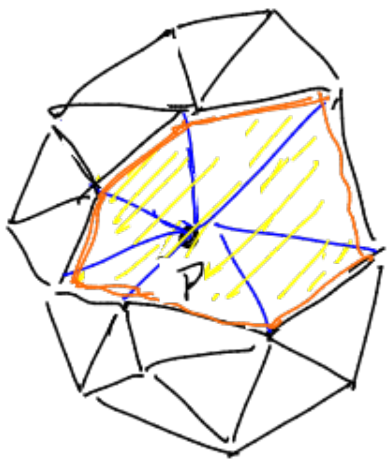


Some region is retriangulated via a star at p.
All the new edges are incident to p.

Correctness. Why is this limited Test and retriangulation sufficient?

- all the tests and flips we do are correct
- the only issue is that we do not test all the edges to check if they are illegal

Correctness. Why is this limited Test and retriangulation sufficient?



Claim. Edges not incident to p are legal.

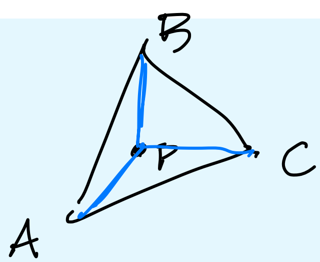
black edges — nothing changed

orange edges — we tested.

blue edges?

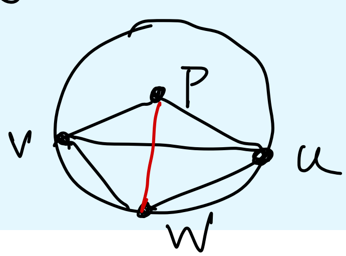
Lemma. Any edge we add (incident to p) is legal. In fact, Delaunay.

– original edges



are Delaunay because Circle(ABC) was empty & can shrink to empty circle through e.g. Ap

– edge created by flip



we flipped because p in Circle(u v w)
p is the only site in there.
shrink to get empty circle through pw
∴ pw is Delaunay.

[Randomized] Incremental Delaunay triangulation algorithm.

Analysis of expected run time when points are inserted in random order.
Note: we are still ignoring how to find which triangle contains p (and its runtime).

Lemma. The expected time to insert one point is O(1).

Proof. Time spent ~~to do~~ Tests and updates when inserting p_i

$= O(\#\text{edges incident to } p_i \text{ after the updates})$

$= O(\text{degree of } p_i)$

So we want expected degree of p_i in $\mathcal{D}(\{p_1 \cdots p_i\})$

$=$ avg. degree in $\mathcal{D}$ — a planar graph — so $O(1)$.

This shows that expected # triangles created over the course of the algorithm is $O(n)$.

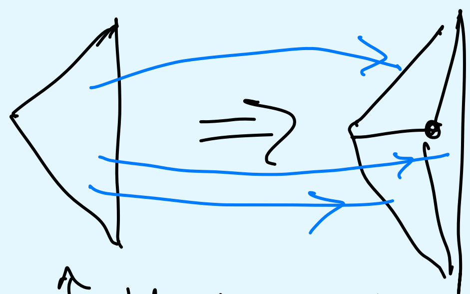
[Randomized] Incremental Delaunay triangulation algorithm.

Final issue: How to find the triangle containing p.
The method is easy, the analysis is not.
Note: it is this part of the algorithm that causes the O(n log n) expected behaviour.

The idea is like Kirkpatrick's Point Location.
Maintain the history of triangles and changes to them. Then "trace" point p_i through the changes.

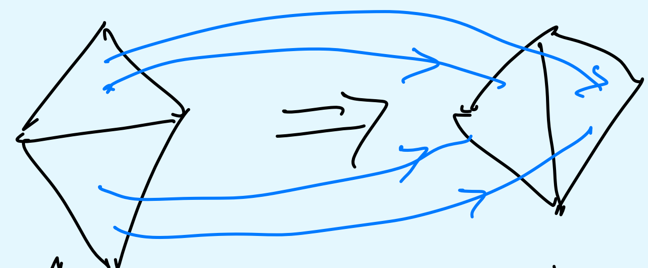
Two possible updates to triangles:

1



old Δ points to 3 new Δs

2



each old Δ points to 2 new Δs

Keep all this info.

<u>Claim</u> Expected size $O(n)$.

because expected # triangles is $O(n)$

[Randomized] Incremental Delaunay triangulation algorithm.

Final issue: How to find the triangle containing p.

How to “trace” p:

- initially (with one big triangle) p is in the big triangle
- at each update, the triangle containing p points to 2 or 3 new ones — check which one contains p

This completes the description of the algorithm.

Analysis of expected work to trace p_i

Can prove it is O(log i). Then total expected work to add all points is

$$O(\sum_i \log i) = O(n \log n)$$

First idea: charge work of tracing p_i to each triangle T in the sequence that contains p_i

Better idea: charge work to Delaunay triangles that appear in the sequence.

Can show that the expected work for triangles of D({p_1, . . . p_j }) is O(3 / j)

$$O(\sum_{j=1}^{i-1} \frac{3}{j}) = O(\log i)$$

Harmonic series

There is a lovely backwards analysis involved. For details, see [CGAA].

What primitive operations are needed for this algorithm?

Given 4 points, A, B, C, D, is D inside Circle(A,B,C)?

Use the mapping from last day

$$(x, y) \longrightarrow (x, y, z = x^2 + y^2)$$

Then the test becomes: is D below the plane through A, B, C?

This is a Sidedness test in 3D, and can be decided with a few multiplications, additions, subtractions.

Summary

- a randomized incremental algorithm for the Delaunay triangulation
- the idea of flipping illegal edges to get to the Delaunay triangulation
- the Delaunay triangulation maximizes the angle vector

References

- [CGAA] Chapter 9.
- [Zurich notes] Chapter 5.