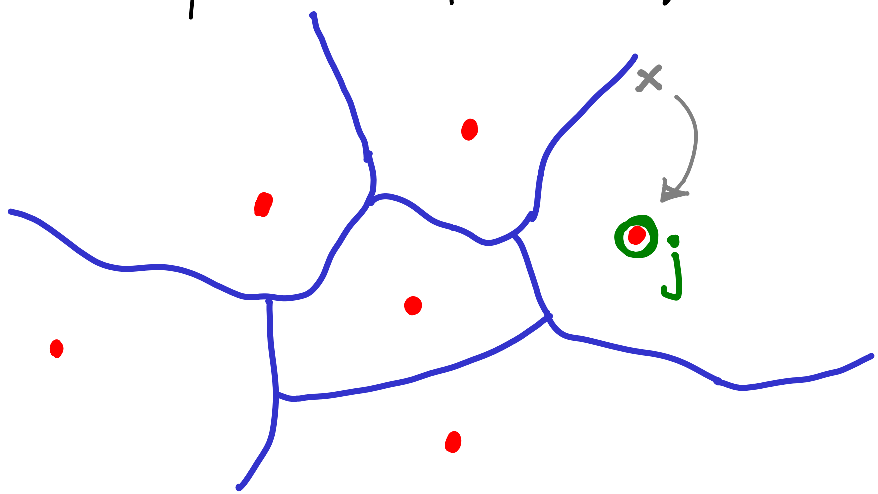
VORONOI DIAGRAMS

INPUT : n points in $\mathbb{R}^2$ $\{p_1, p_2, \ldots\ldots p_n\}$

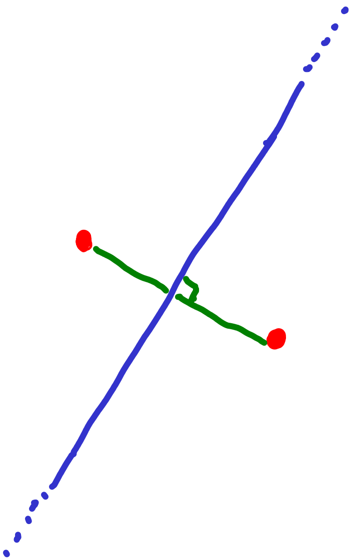
OUTPUT : a partition of $\mathbb{R}^2$ into <u>cells</u> s.t.
any point inside cell j is closer to p_j than
to any other p_i $(i \neq j)$.

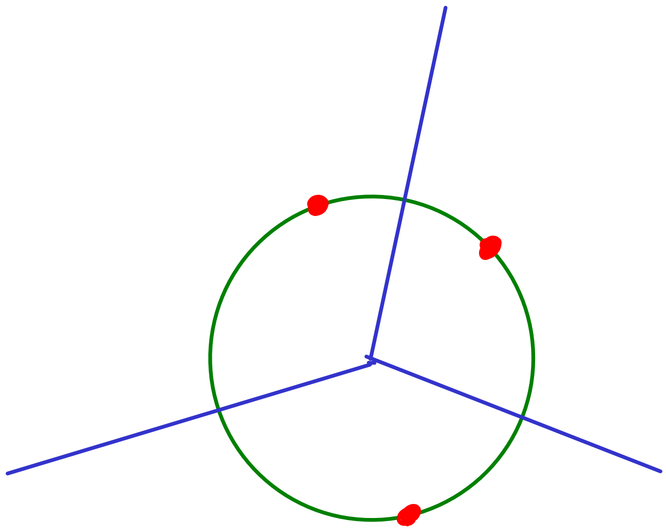


PROPERTIES OF THE VORONOI DIAGRAM

SHAPE of cells (i.e. their borders)

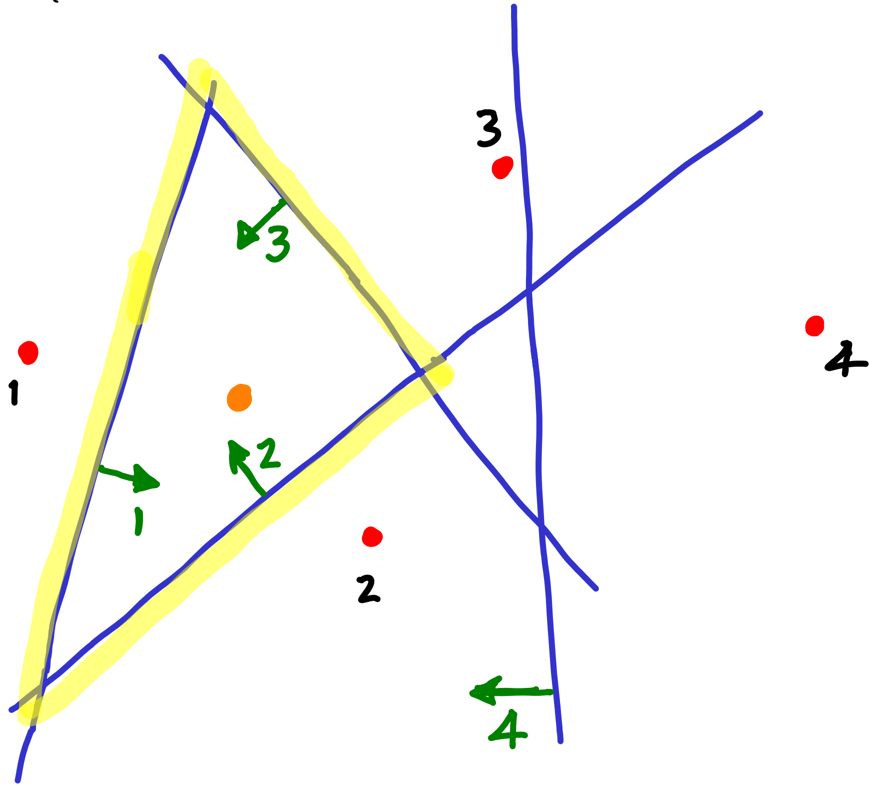
↳ curvature? vertex degree? length? convexity?



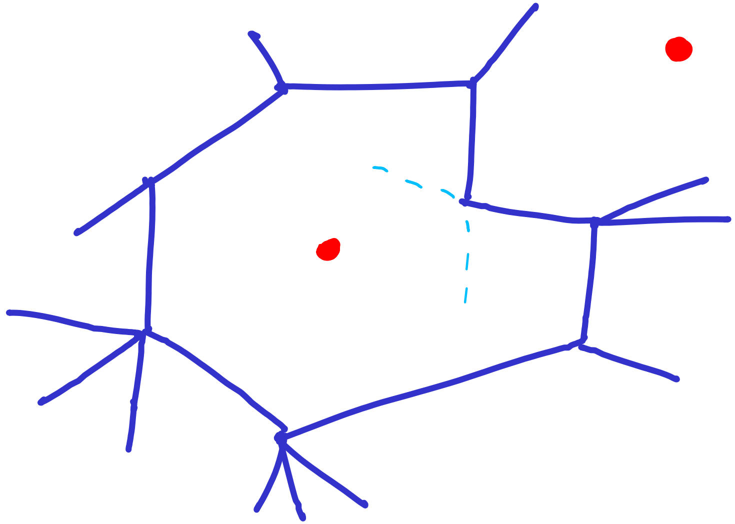


- VORONOI EDGES ARE STRAIGHT
  ↳ Each cell : $\leq n$ bisectors

- CONVEX : Each cell : intersection of halfplanes.



... OR BY CONTRADICTION

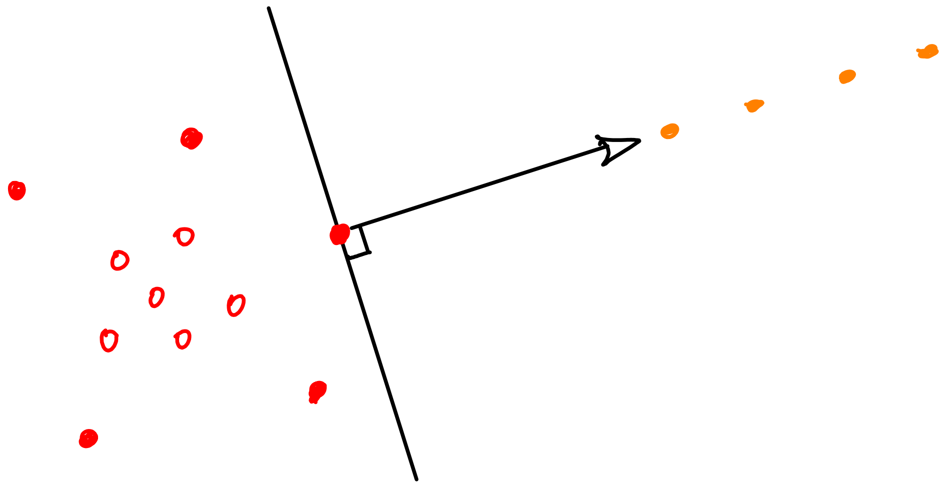


NOTICE : EXACTLY 1 EDGE BETWEEN SITES

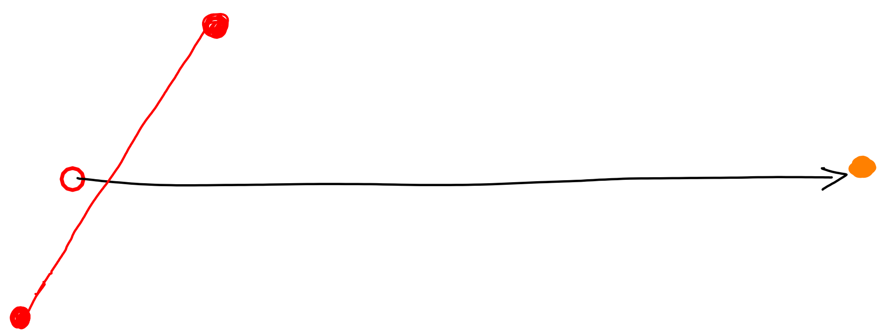
- VORONOI EDGES CAN BE INFINITELY LONG

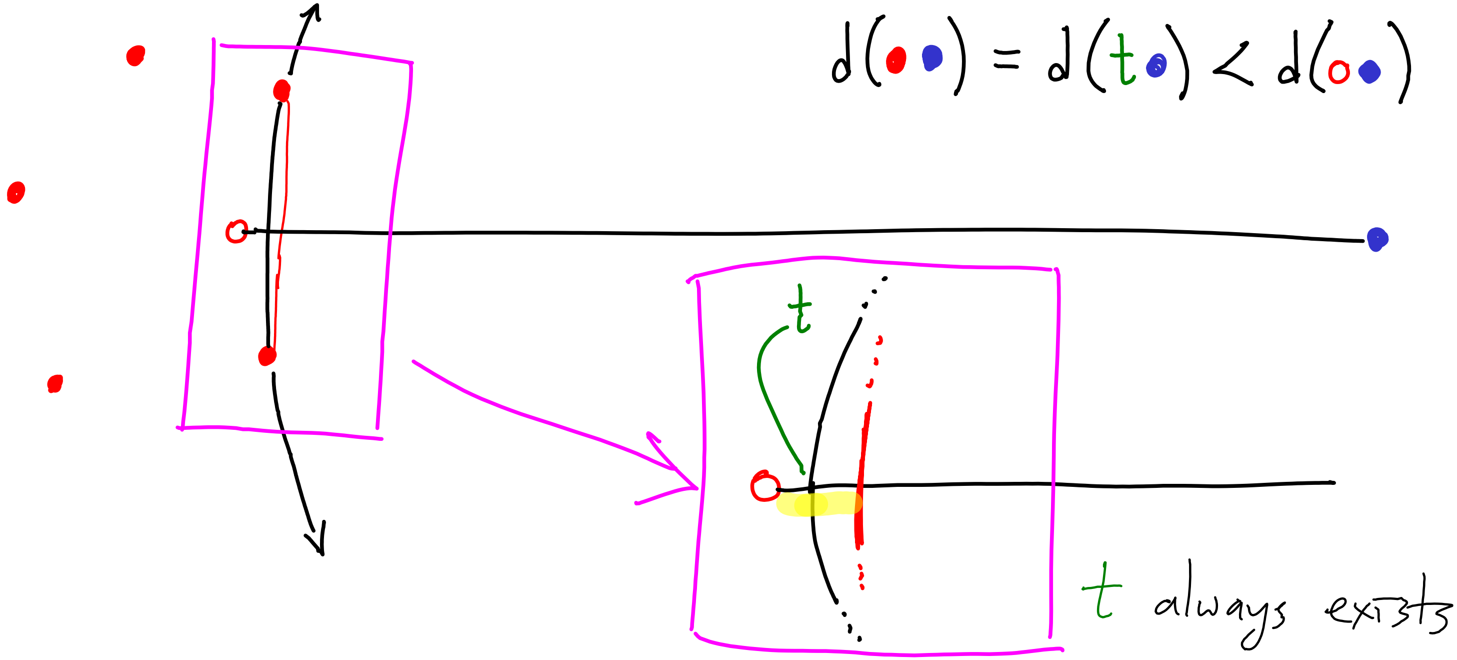
↳ for which Voronoi sites ?

ANY CONVEX HULL VERTEX HAS AN INFINITE CELL

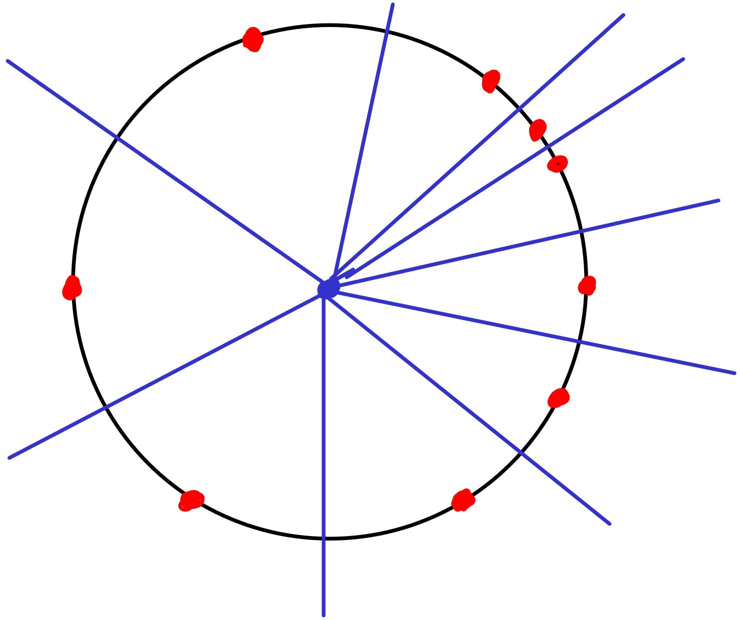


CAN AN INTERIOR POINT HAVE ONE?





- VERTEX DEGREE ?



HIGH DEGREE = CO-CIRCULAR POINTS

ASSUME GENERAL POSITION

(only for explanation)

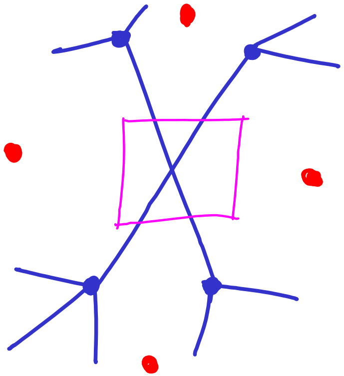
$d=3$

- SIZE OF CELL CAN STILL BE n.
  #CELLS $= n$

SIZE OF GRAPH ....

THE VORONOI DIAGRAM IS PLANAR.

↳ PLANE



$O(n)$ edges (+ vertices)

$\quad\quad e \quad\quad\quad v$

@∞

$V - E + F = 2$ / $F = n$ / $V = v + 1$ / $2e \geq 3(v+1)$

WHY IS THE VORONOI DIAGRAM USEFUL ?

- IN GEOMETRY : NEAREST NEIGHBOUR CLASSIFICATION

  ↳ by point location , $O(\log n)$

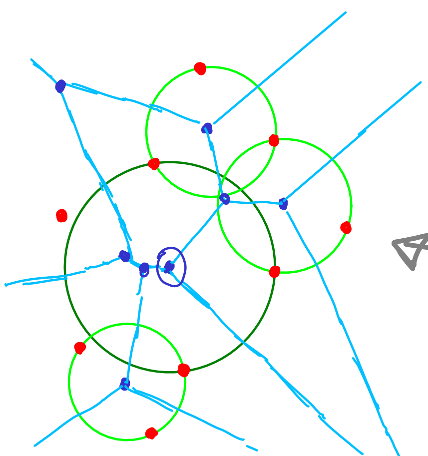
  (competitive)

  FACILITY LOCATION (also collision avoidance)

  ↳ Largest empty circle (next page)

  SHAPE EXTRACTION

  ↳ MEDIAL AXIS (will see later)

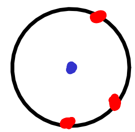


LARGEST EMPTY CIRCLE (LEC)

[x = VORONOI VERTEX] $\Longleftrightarrow$ [LEC(x) has ≥ 3 points on it]

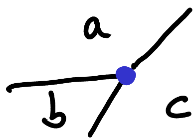
$= 3$ for g.p.

1) $\Longleftarrow$ ... easy :



x belongs to all 3 sites
↳ hence vertex

2) $\Longrightarrow$



: any vertex x belongs to 3 cells, a, b, c.
so $d(xa) = d(xb) = d(xc)$
& $\nexists y$ s.t. $d(xy) < d(xa)$ ... QED

In fact, <u>any</u> point on Vor. Diag. is defined by a LEC, of ≥ 2 pts.

↳ collision avoidance

THE GLOBAL LEC MUST BE CONSTRAINED BY 3 PTS.

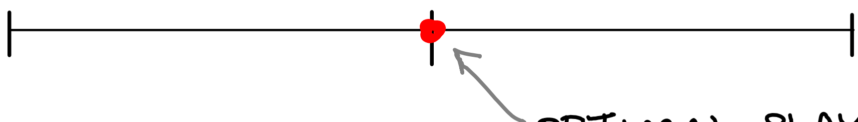
- OTHER USES : MANY! SEE LINKS AFTER CLASS.
  - The Voronoi game - next page
  - Art ... see David Eppstein

IN NATURE

Hex lattice, crystal growth, plant/root growth, breadfruit!

THE VORONOI GAME

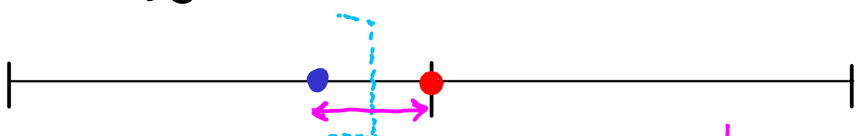
- 2 PLAYERS, k ROUNDS – WINNER IS PLAYER with MOST VORONOI AREA (within a closed region)
- LOOK AT 1D FIRST .... with 1 ROUND



OPTIMAL PLAY BY ●

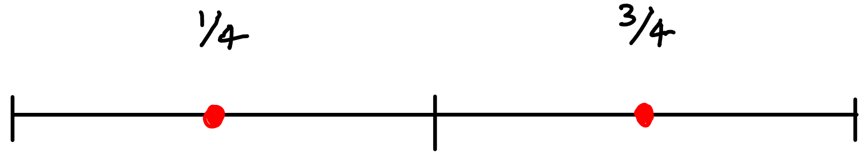
● $\frac{1}{2}+\varepsilon$
● $\frac{1}{2}-\varepsilon$

- NEXT: 2 ROUNDS: SUPPOSE ● PLAYS at CENTER AGAIN.



– THEN ● PLAYS AT SOME FIXED DISTANCE d, NOT TOO FAR AWAY.
– IF ● DOESN'T PROTECT RIGHT HALF, THEN ● WILL STEAL $\frac{1}{2}-\varepsilon$ BUT ALSO AT LEAST $d/2$ ON THE LEFT $\Rightarrow$ ● WINS.
– SO ● MUST PLAY ON RIGHT SIDE. THUS ● HAS $\frac{1}{2}-\frac{d}{2}$ ON LEFT, BUT ALSO WILL EASILY KEEP $>d/2$ ON RIGHT.

SO, PLAYER 1 WINS 1 ROUND BUT LOSES w/ SAME STARTING STRATEGY ON 2 ROUNDS. CAN PL.1 WIN?



Let player 1 reveal both moves at start

a) 2 • between ⇝ < 1/2
b) 2 • outside ⇝ < 1/2
c) 1+1 .... same!

PLAYER 1 WINS

[is there another winning strategy?]

[what if w/o revealing?]

Try shifting solution, or changing gap size

• must play at 3/4 if • plays 1/4.
THEN WHAT?

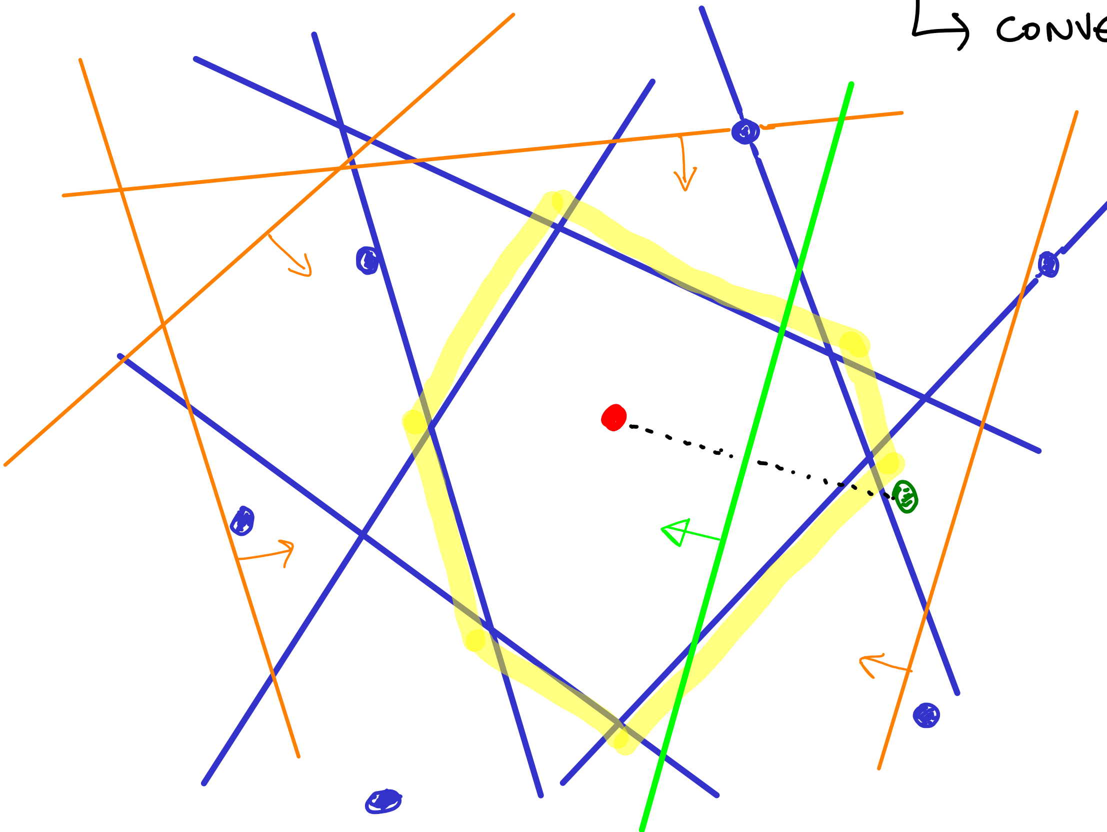
• 2D: NO THEORY FOR TODAY. SEE APPLETS!

COMPUTATION

- BRUTE FORCE – (bisector intersection)

↳ FOR EACH POINT : FIND CELL

↳ CONVEX POLYGON + HALFPLANE INTERSECTION



- CELL SIZE : $O(n)$
- INTERSECT : $O(\log n)$
- $O(n)$ halfplanes

TOTAL $O(n \log n)$ per cell

even if cell ends up very small

YOUR HOMEWORK

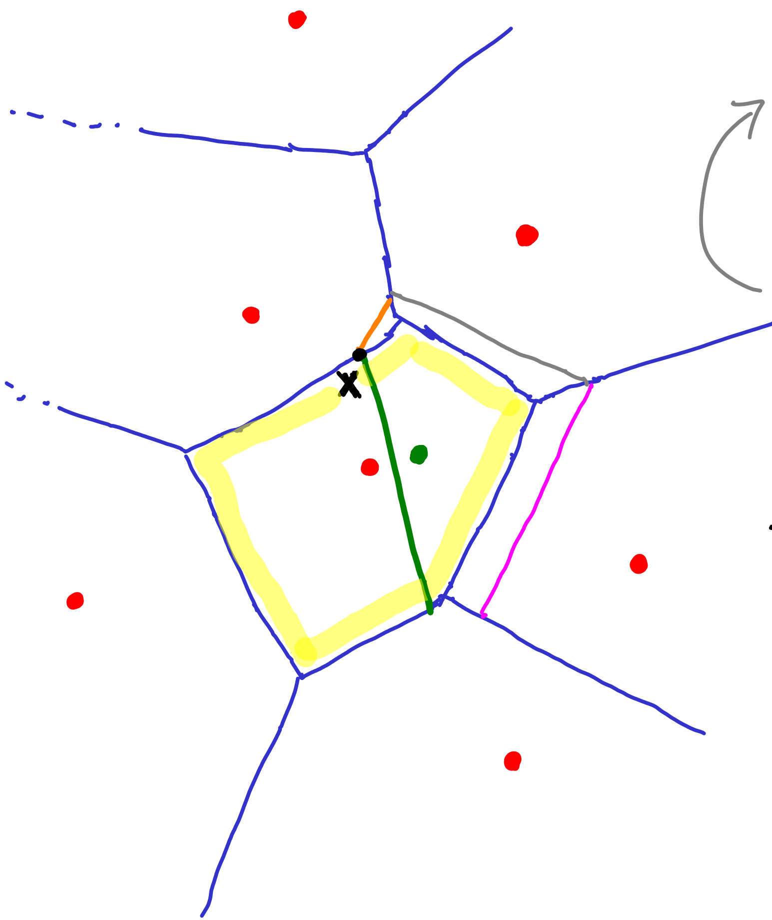
1) CONSTRUCT THE VORONOI DIAGRAM
(BRUTE FORCE IS OK)
(YOU CAN USE LAST WEEK'S CODE)

2) ALLOW USER INPUT

For (1) use ~15 fixed pts.

For (2) see link to a sample applet on todays webpage.
Use evt.id
Feel free to copy code (from me)

• INCREMENTAL – (new point "steals" its region)



1) POINT LOCATION FOR

2) COMPUTE BISECTOR IN ITS CELL

3) FIND X = INTERSECTION $\cap$

4) REPEAT IN NEW CELLS

↳ BISECTOR, INTERSECTION...

↳ BISECTOR, BISECTOR, etc

TIME PER INSERTION?

$$\left.\begin{aligned} &1)\ O(\log n)\\ &2)\ O(1)\\ &3)\ O(\log n)\end{aligned}\right\} \times \#\text{cells} : O(n)$$

TOTAL PER INSERTION $= O(n \log n)$?!

IN FACT: FORM NEW CELL BY WALKING ON OLD GRAPH. (don't even bother w/ $O(\log n)$ for step 3)

actually step 3 $\sum O(\log c_i)$ $[\sum c_i = O(n)]$

↳ STEP 3 $= O(c)$: $c =$ size of cell

↳ $\sum (\text{step3})_i = \sum O(c_i) = O(n)$ per point

↳ TOTAL $= \boxed{O(n^2)}$

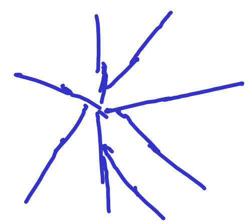
ON AVERAGE, FOR RANDOM POINTS $= O(n)$!

CAN WE DO FASTER?

• YES... but first: LOWER BOUND?

↳ easy .... $\Omega(n \log n)$

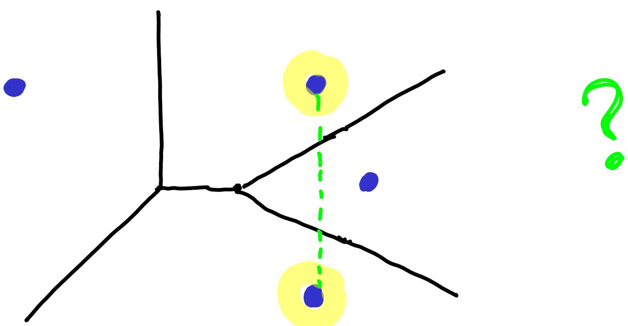
WHY? CONVEX HULL



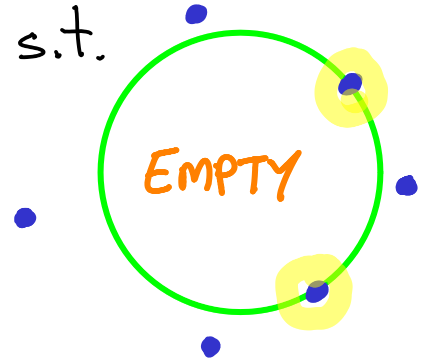
let's delay $O(n \log n)$ some more...

DELAUNAY TRIANGULATIONS

- THE DUAL OF VORONOI (vertex $\leftrightarrow$ face)
  - WHY IS IT A TRIANGULATION?
    Voronoi vertex : degree 3
  - WHICH PAIRS OF POINTS ARE JOINED w/ AN EDGE?



- IF 2 POINTS HAVE SOME CIRCLE s.t.



THEY SHARE AN EDGE

↳ ANY 2 POINTS w/ ADJACENT VORONOI CELLS.

↳ ANY 3 POINTS CREATING A VORONOI VERTEX

↳ DELAUNAY TRIANGLE

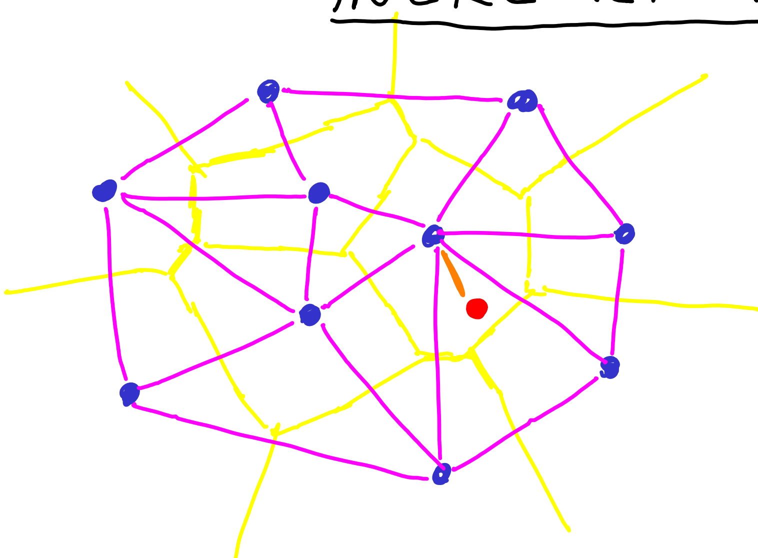
- THE CIRCUMCIRCLE OF A D. TRIANGLE IS EMPTY
  & ITS CENTER IS A V. VERTEX

COMPUTING THE D.T.

- EASY: CONSTRUCT VORONOI and MAKE DUAL

- OTHER WAYS:
  ↳ WHAT IS EQUIVALENT TO THE VORONOI ALGO?

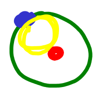
INCREMENTAL



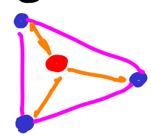
NEW POINT

DEFINITELY CONNECTS TO THE "HOST CELL" SITE

↳ WHY?

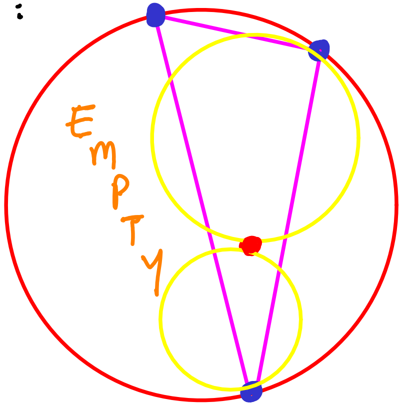


DOES IT CONNECT TO ALL 3 ? TRIANGLE VERTICES

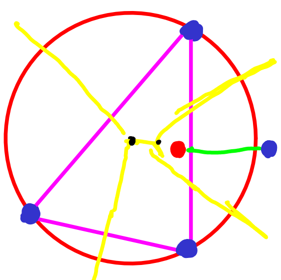


?

RECALL :



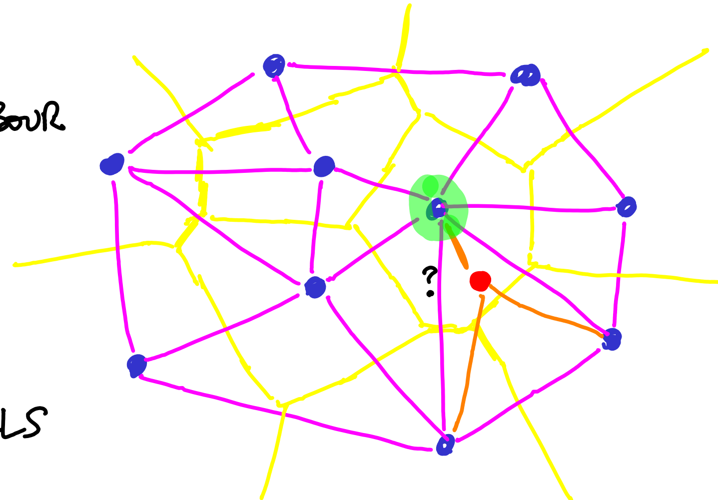
DID I CHEAT ABOVE?
— IS THE HOST CELL SITE
ONE OF THE 3 D.TRIANGLE VERTICES ?



I CHEATED.
BEWARE OF
"PROOF BY PICTURE"

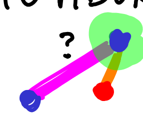
IN FACT, ● NEED NOT BE IN A TRIANGLE!

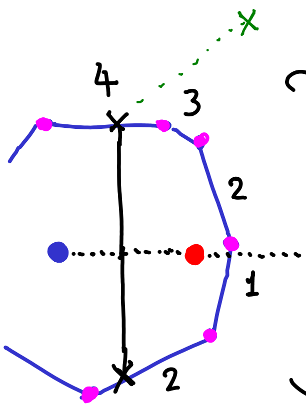
MODIFY INCREMENTAL:
INSTEAD OF NEAREST NEIGHBOUR
START w/ ANOTHER KNOWN NEIGHBOUR



↳ THE 3 NEIGHBORS WILL
BE PART OF A CYCLIC
SEQUENCE OF VORONOI CELLS
AROUND ●

↳ IDENTIFY OTHER NEIGHBORS, IF ANY.

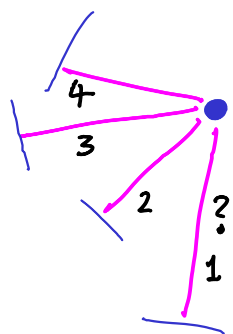
↳ TEST IF  MUST BE DESTROYED.



RECALL IN VORONOI
TO FIND X,
YOU COULD
WALK ON
CELL(●)

↳ THEN CONTINUE
IN NEXT CELL

THIS IS ~ TESTING



REMOVING
A CELL EDGE

=

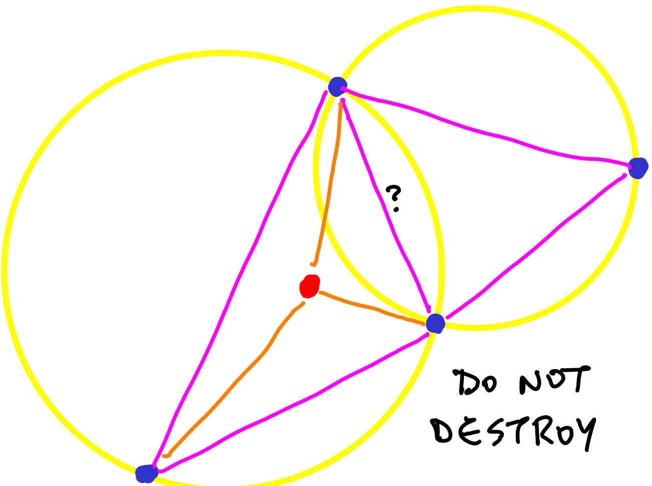
REMOVING
A DELAUNAY
EDGE

DUAL

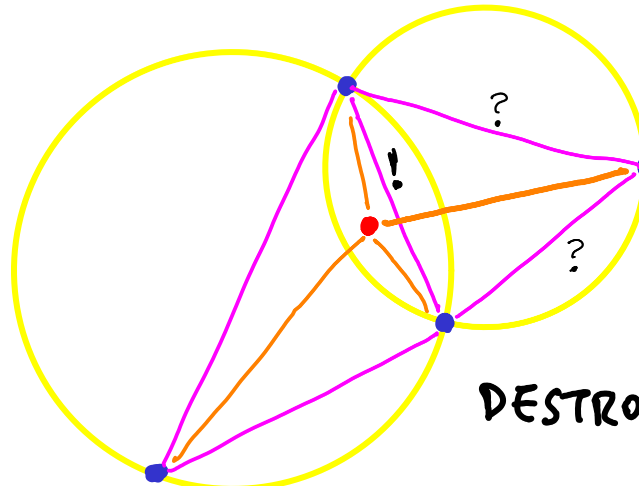
ACTUALLY YOU CAN TEST IN A DIFFERENT ORDER.

DO EMPTY CIRCLE TEST ON ADJACENT

- IF SURVIVES, STOP
- IF NOT, <u>FLIP</u>

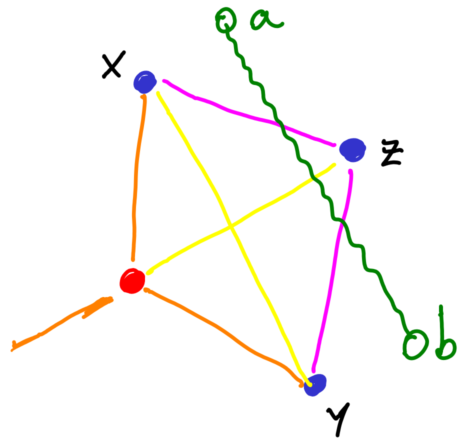


DO NOT DESTROY



DESTROY!

• PROVE THAT NEW EDGE IS GOOD.



i) $\overline{XY}$ NOT GOOD (DESTROYED)

ii) $\overline{\bullet Z}$ NOT GOOD (ASSUME)

SOME NEW EDGE MUST CROSS ΔXYZ

CONTRA-DICTION

→ SO ∃ a,b THAT HAVE AN EMPTY CIRCLE

↳ BUT THEN THAT EDGE SHOULD NOT BE NEW.

in other words any new edge involves •

.... but see previous page too. Shrink LEC(••) : new edge.

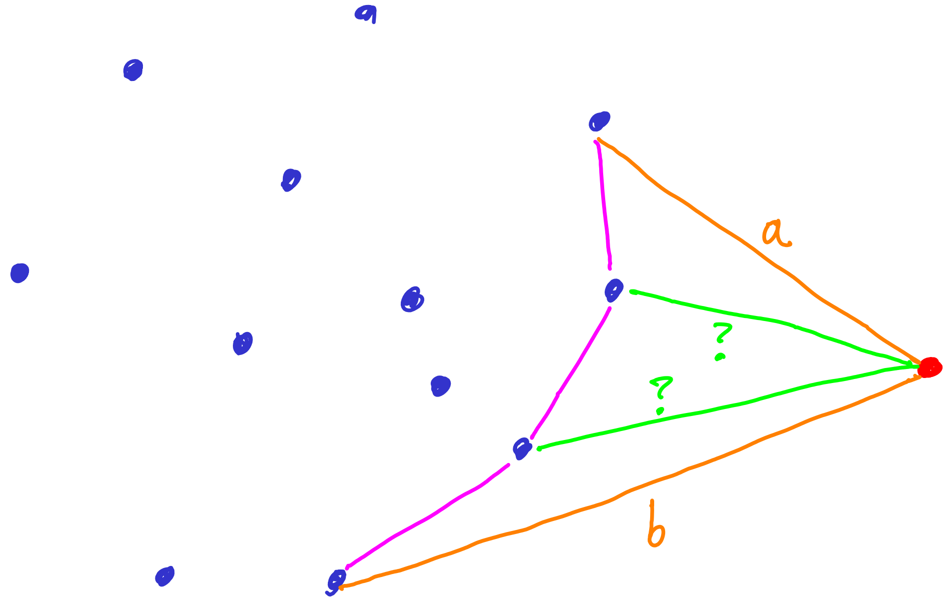
• SO, KEEP BRANCHING & FLIPPING ... $O(n)$

#FLIPS ← PER INSERT.

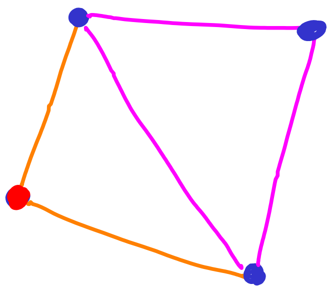
.... $O(n^2)$ but $O(n)$ EXPECTED!

INSERTION OUTSIDE C.H.

a, b : tangent
(will be in DT)



MORE ABOUT THE GEOMETRY OF D.T. & FLIPPING



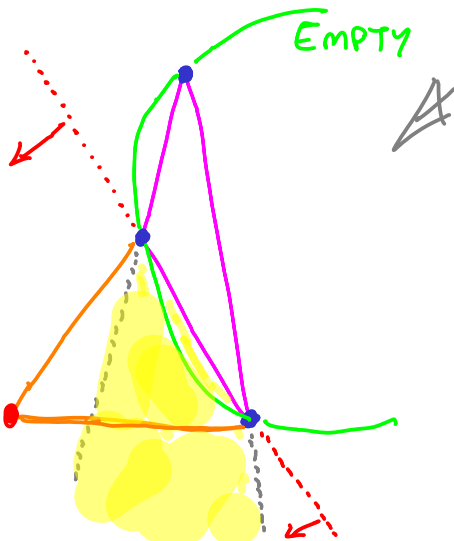
: GUARANTEED D. EDGES

: NOT VERIFIED

REMEMBER, THE QUESTION IS:

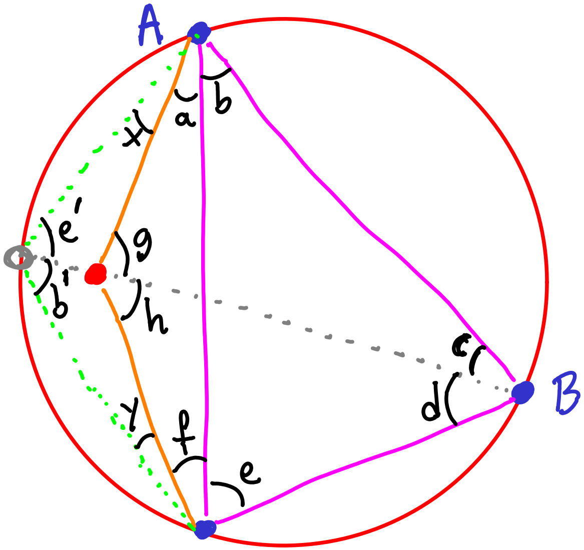
IS △ STILL LEGAL AFTER INSERTING • ?

— IF THIS IS NOT CONVEX, THEN <u>YES</u>



— OTHERWISE, FOR CONVEX QUADRILATERAL EXACTLY ONE DIAGONAL SURVIVES.

↳ we have seen how to determine



SUPPOSE ● IS INSIDE CIRCUMCIRCLE ▽

EXTEND ● - - - ● NEW DIAG.

D.T.1 : A●B ‖ N.D.1 : A●C

D.T.2 : B●C ‖ N.D.2 : ABC

↖ not Del.

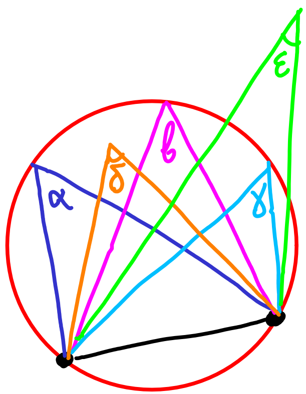
Min angle in $D.T._i$:

$\min\{h, g, (a+b), d, c, (e+f)\}$

Min angle in $N.D._i$:

$\min\{b, e, (g+h), a, f, (c+d)\}$

$\hat{\alpha} = \hat{\beta} = \hat{\gamma} \left\{ \begin{array}{l} < \hat{\delta} \\ > \hat{\varepsilon} \end{array} \right.$



⟷ FROM THALES : $b = b'$ & $e = e'$

↳ $h > b$ & $g > e$

ALSO : $d = a + x$ & $c = f + y$

↳ $d > a$ $\quad$ $c > f$

D.T. MAXIMIZES MINIMUM ANGLE OVER <u>ALL</u> TRIANGULATIONS

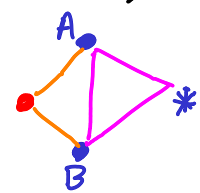
If minDT is d, c, h OR g ⇒ minND < minDT

If minDT = $a+b$, minND ≤ b < minDT. Same for $(e+f)$.

HOMEWORK

- DEMONSTRATE DELAUNAY INCREMENTAL (FLIPPING)
  - STORE TRIANGLES (array is ok; assume max ≈ 30)
  - POINT LOCATION: brute force ok
  - COLOR NEW EDGES
  - FIND ADJACENT TRIANGLES (brute force ok)

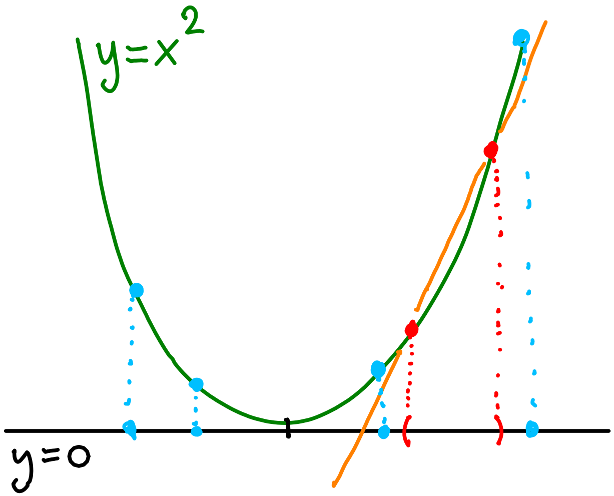
    search AB*



  - DO CIRCUMCIRCLE TEST: use bisector intersection

- YOU WILL NEED TO STORE "UNFINISHED" TRIANGLES
- USE COLOR AS MUCH AS POSSIBLE

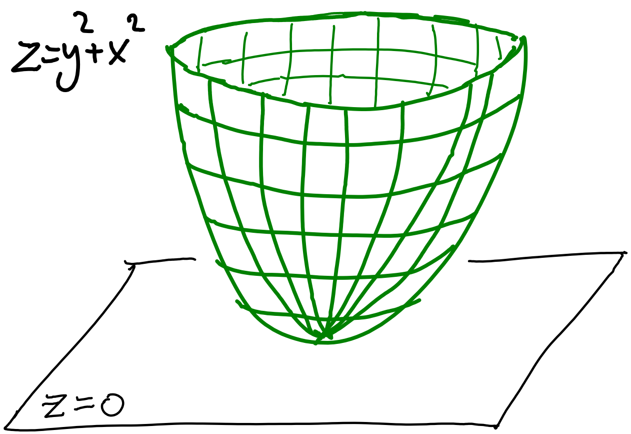
CONSTRUCTING DELAUNAY in $O(n \log n)$

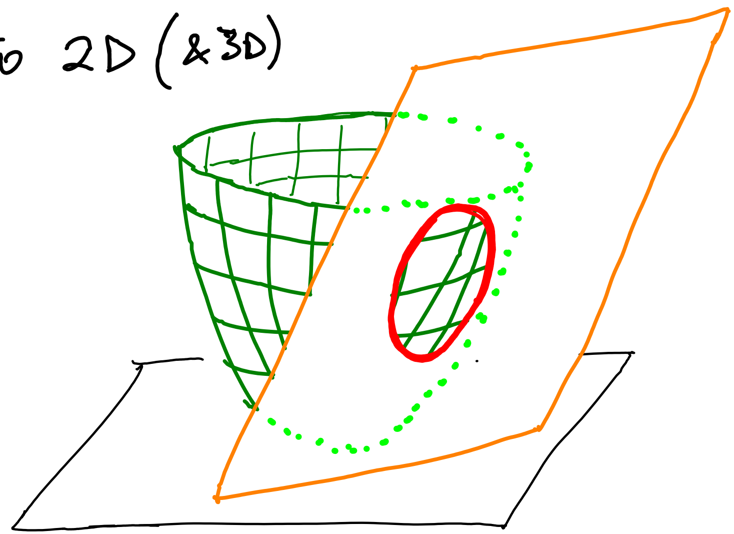


In 1D the interval ( ) is empty iff the endpoints raised onto $y=x^2$ are on the lower convex hull. ex: ●

(any convex function works)

BACK TO 2D (& 3D)

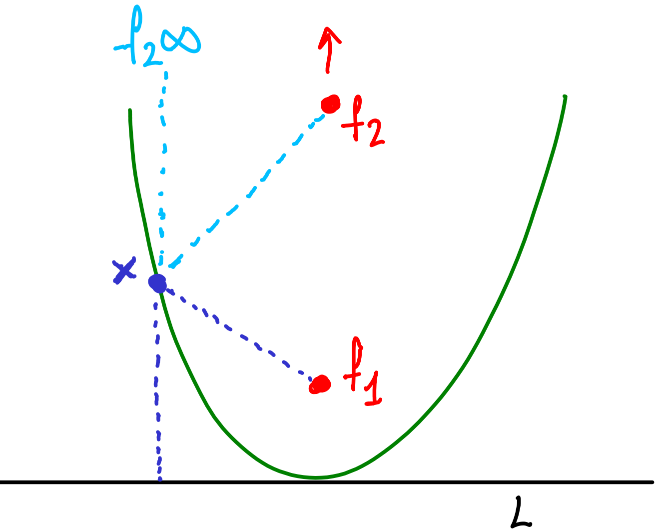




- THE INTERSECTION OF A PARABOLOID WITH ANY NON-VERTICAL PLANE IS AN ELLIPSE.

(see notes by VERA SACRISTAN)

- intuition: ANY PARABOLOID is ~ limit of ELLIPSOID w/ ONE FOCAL POINT AT ∞



ELLIPSE: $d(x,f_1) + d(x,f_2) = c$

As $f_2 \to \infty$ ... $d(x,f_2) \sim d(f_2,L) - d(x,L)$

$\Downarrow$

PARABOLA: $d(x,L) = d(x,f_1)$ $\Rightarrow c \sim d(f_2,L)$

So,
PLANE $\cap$ SPHERE = CIRCLE
PLANE $\cap$ ELLIPSOID = ELLIPSE
PLANE $\cap$ PARABOLOID = ELLIPSE

Note: the above is ...lim, but the intersection is actually exact.

In FACT the ELLIPSE PROJECTS VERTICALLY TO A CIRCLE...

IF we USE $z = x^2 + y^2$
(see notes provided online)

- SO IF YOU HAVE POINTS INSIDE A horizontal CIRCLE C on $z=0$
  AND YOU LIFT THEM TO $z = x^2 + y^2$
  THEY WILL BE <u>UNDER</u> THE CORRESPONDING CUTTING PLANE P
  that is DEFINED by the LIFTING of C → an ellipse

- SO IF 3 POINTS ARE <u>ON</u> AN EMPTY CIRCLE C
  THEN THERE IS NO OTHER POINT BELOW P
  $\Rightarrow$ THE 3 POINTS LIFTED ARE ON THE 3D CONVEX HULL

- ANY FACE F OF THE CONVEX HULL: $\downarrow$ EMPTY CIRCLE
  w/ VERTICES OF F ON THE CIRCLE.

CONCLUSION:

- COMPUTE CONVEX HULL of LIFTED POINTS.
  lower
- (for general position every face is a triangle)
- PROJECT DOWN & GET DELAUNAY TRIANGULATION

BONUS: 3D CONVEX HULL IS $\underline{O(n \log n)}$

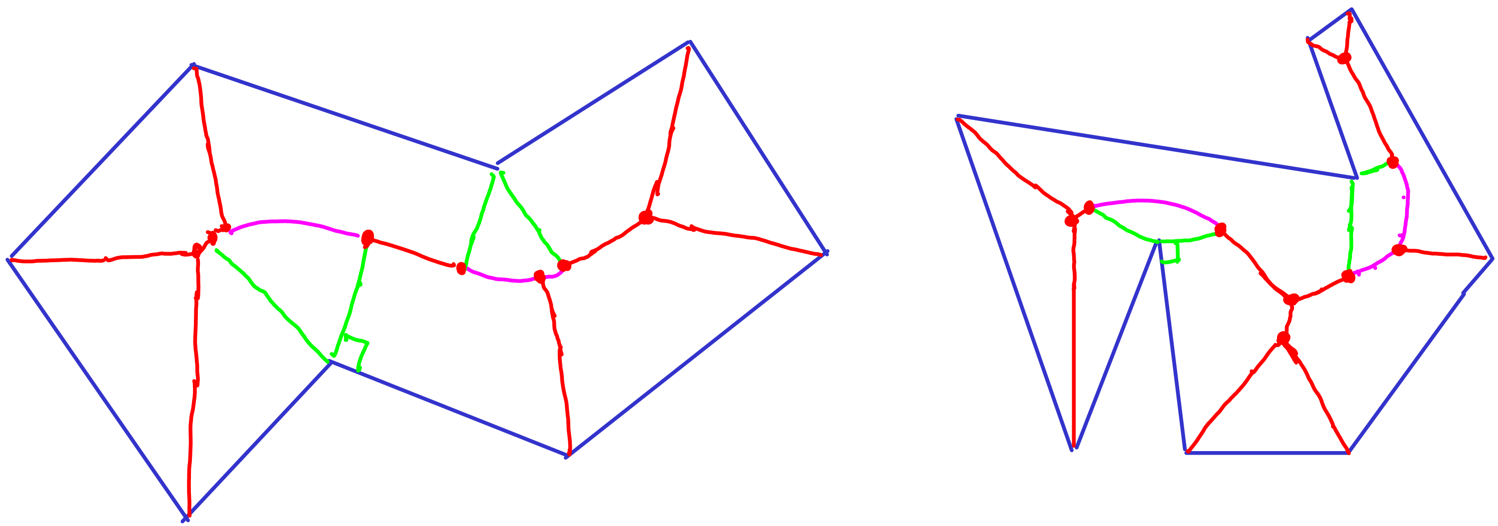
- OTHER $O(n \log n)$ ALGORITHMS FOR VORONOI/DELAUNAY
  - FORTUNE'S SWEEP
  - DIVIDE & CONQUER

  } WILL COVER IF WE HAVE TIME

MEDIAL AXIS

- ~ VORONOI DIAGRAM OF A POLYGON

  ↳ same questions : SHAPE? SIZE?



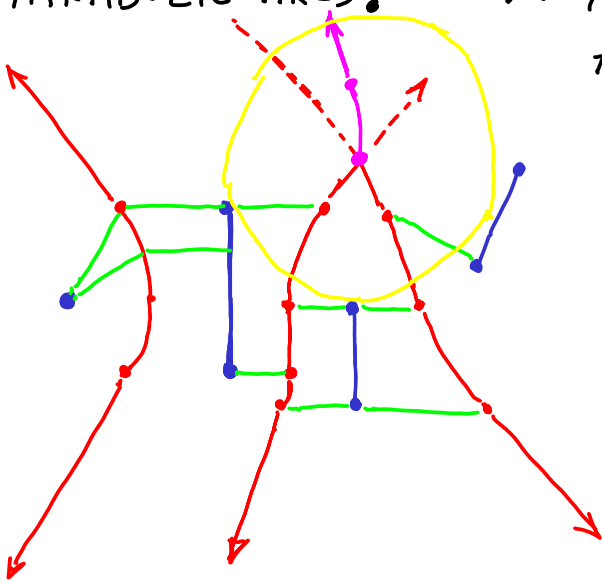
not part of M.A.

- SEGMENTS : CONVEX ANGLE BISECTORS & ANY POSITION EQUIDISTANT TO TWO <u>EDGES</u>

- VERTICES : POINTS EQUIDISTANT TO ≥ 3 POSITIONS ON BOUNDARY

  let's assume $= 3$ for now

- PARABOLIC ARCS! : ANY POSITION EQUIDISTANT TO AN EDGE AND A (REFLEX) VERTEX.

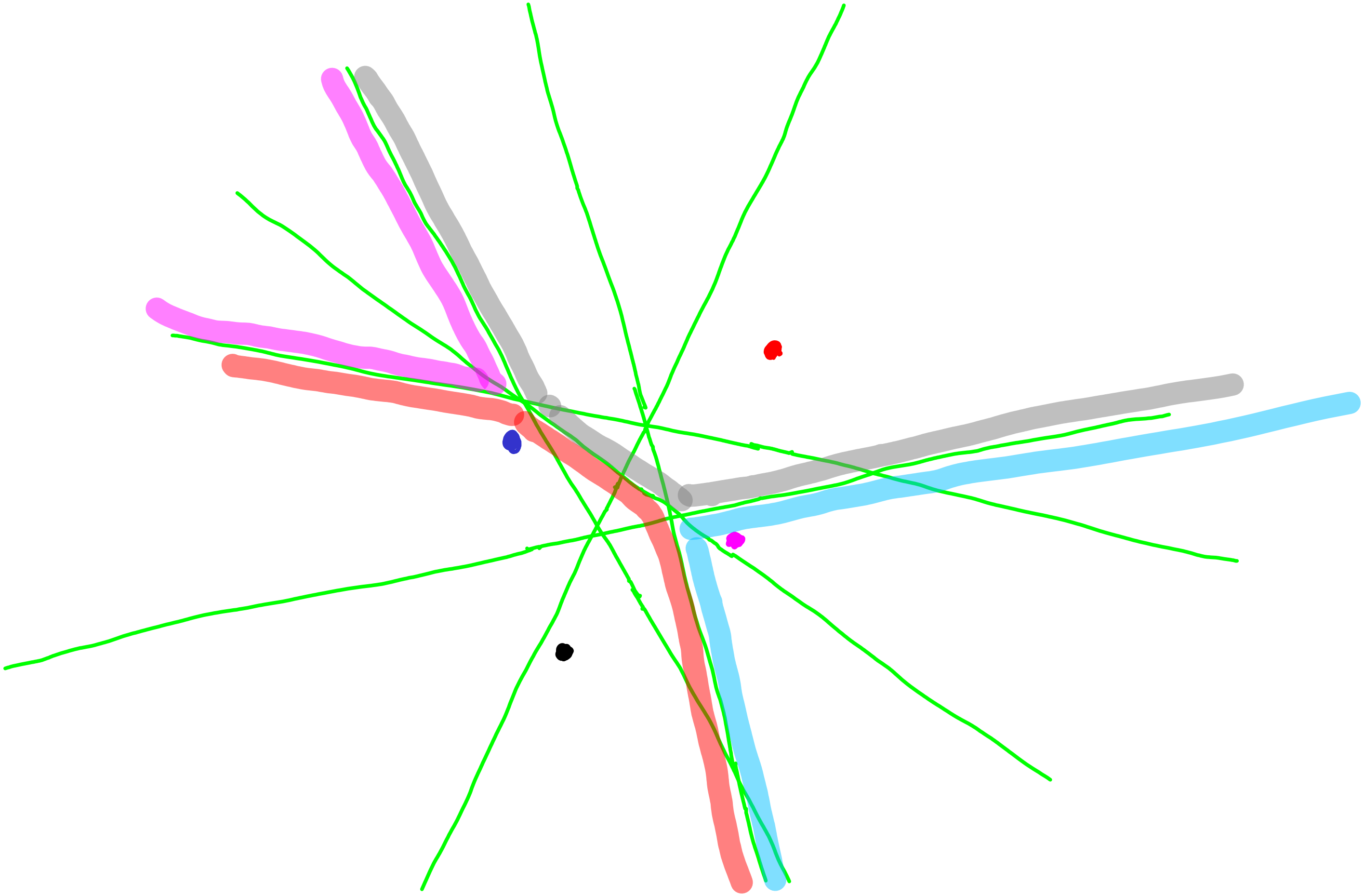


- COMPUTATION : $O(n)$ for polygons

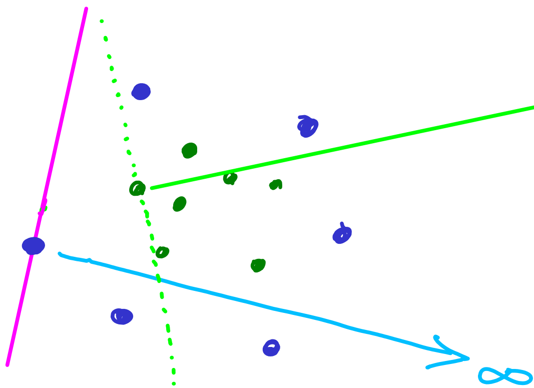
  ↳ beyond scope of this class

↖ think of collision avoidance

FURTHEST POINT VORONOI DIAGRAM



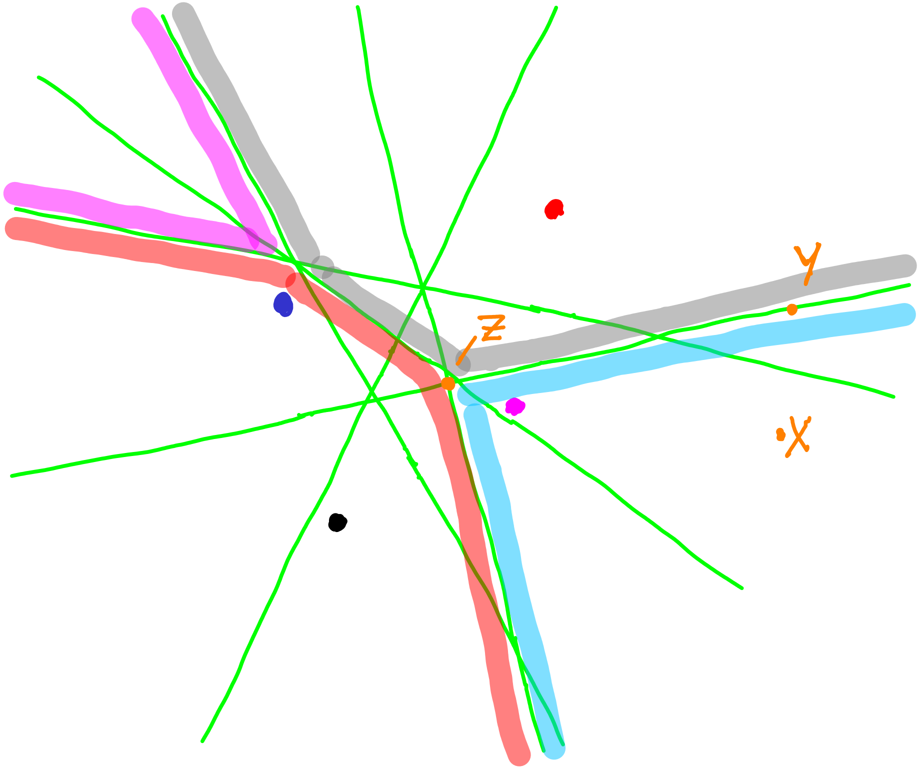
- WHICH POINTS HAVE ∞ REGIONS?
  WHICH HAVE BOUNDED REGIONS?



ONLY C.H. point CAN HAVE ∞ CELL

... BUT ∞ CAN BE MADE ε

- ONLY C.H. POINTS HAVE CELLS.
- F.V.D. IS A TREE.



X : is uniquely furthest

Y : equally far

Z : at center of circle with on it and all other points INSIDE

SMALLEST ENCLOSING CIRCLE

$O(n \log n)$

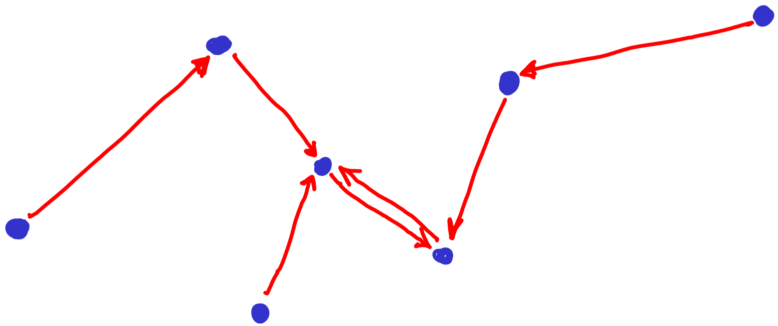
OTHER METRICS

- RECALL THAT EUCLIDEAN VORONOI CELLS CAN BE "GROWN" BY EXPANDING CIRCLES
- WHAT ABOUT L_1? L_∞? etc?!
  - how do we grow cells?

↳ see links (taxi cab geometry, "different metrics")

PROXIMITY GRAPHS

- NNG : NEAREST NEIGHBOUR GRAPH
  - FOR EACH POINT : CONNECT TO CLOSEST POINT.

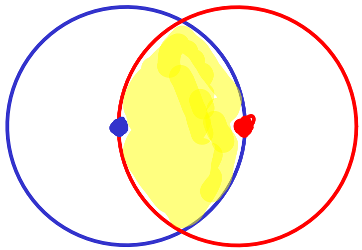


PROPERTIES: CONNECTED ?
#EDGES : n (w/o double counting?)

- ANY EDGE in NNG MUST BE IN MST
  (PROVE BY CONTRADICTION)
  ↳ COMPUTE NNG in $O(n)$ after MST.
- $\Omega(n \log n)$ (EVEN FOR CLOSEST PAIR PROBLEM)
  element uniqueness

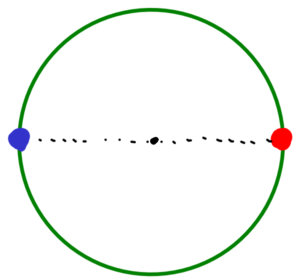
NOTE : k-NNG also exists

RNG : RELATIVE NEIGHBORHOOD GRAPH



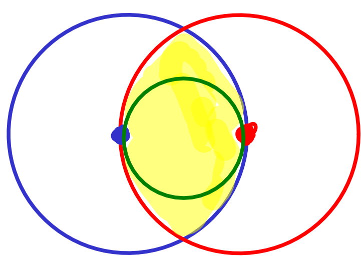
- JOIN ●—● IF LUNE IS EMPTY.

GG : GABRIEL GRAPH



- JOIN ●—● IF ○ IS EMPTY.

WHICH OF THE 2 HAS MORE EDGES?

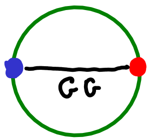


RNG : MORE RESTRICTIVE
↳ FEWER EDGES

$RNG \subseteq GG$

BUT HOW MANY EDGES? $f(n)$

ANY EDGE IN GG MUST ALSO BE IN D.T.



is an empty circle, are Voronoi neighbors.

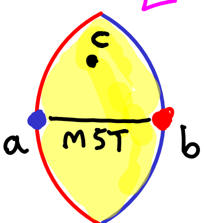
↳ GG $\subseteq$ DT $\Rightarrow$ $O(n)$ edges.

and clearly the opposite is not true.

• CLAIM: MST $\subseteq$ RNG

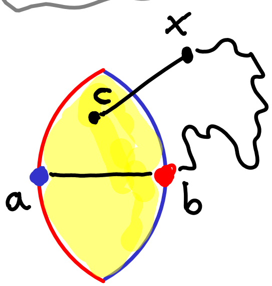
↳ suppose not true : $\overline{ab}$ in MST but not in RNG.

Then $\exists$ c in LUNE.



Now, fact: $\left.\begin{array}{l}\overline{ac} < \overline{ab} \\ \overline{bc} < \overline{ab}\end{array}\right\}$ So $\overline{ac}$ OR $\overline{bc}$ each cannot be in MST.

(we could replace $\overline{ab}$ in MST)



So $\exists$ path from c to some x & then to

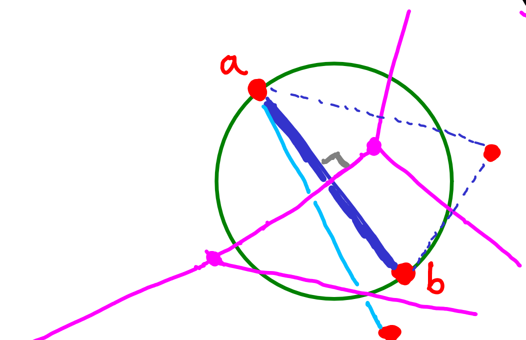
- If is a leaf, use $\overline{ac}$ instead of $\overline{ab}$.
  (symmetric for)
- If neither a leaf, replace either.

CONCLUSION: MST can be IMPROVED!

THUS NNG $\subseteq$ MST $\subseteq$ RNG $\subseteq$ GG $\subseteq$ DT

ALGORITHMS :

GG $\rightarrow$ KEEP EDGE $\overline{a,b}$ OF DT IF IT CROSSES THE VORONOI DIAGRAM AT CELL(a) $\cap$ CELL(b)



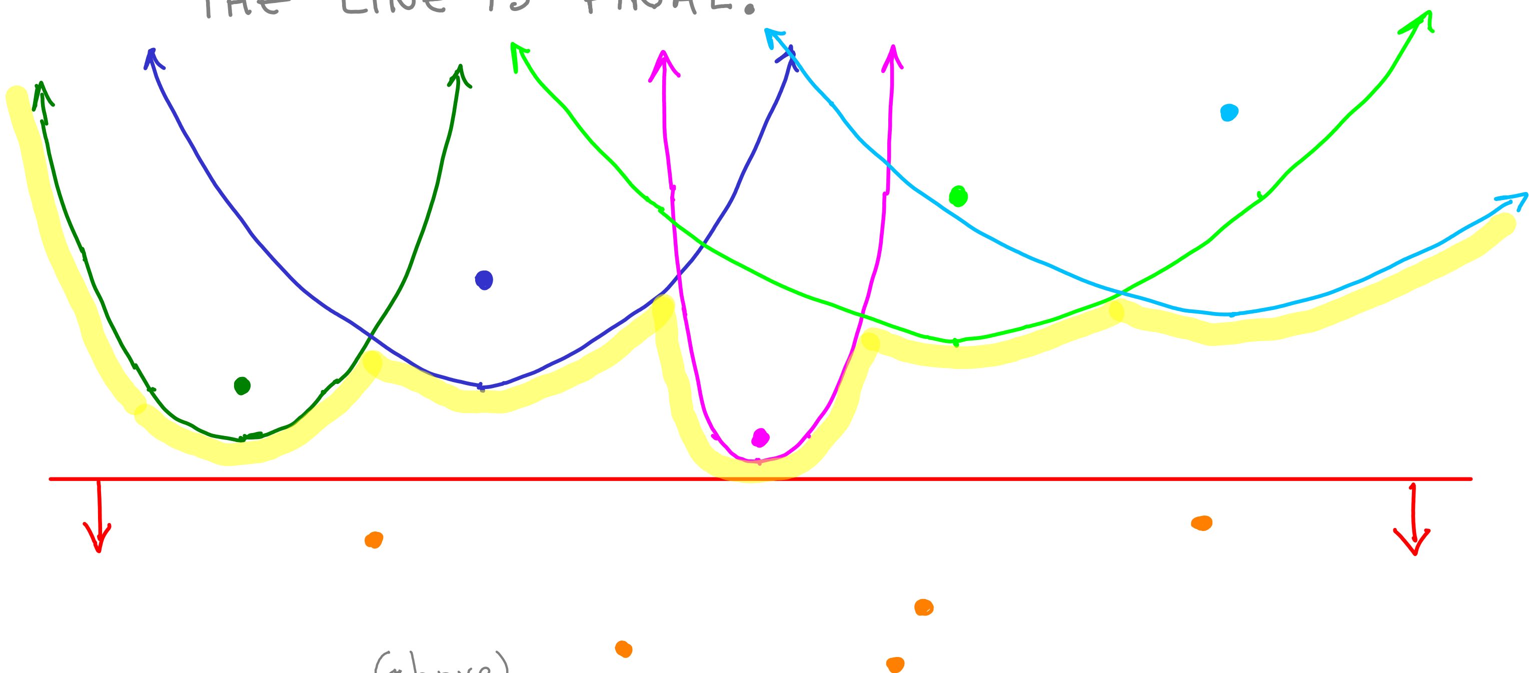
O(n) after D.T.

- RNG CAN ALSO BE COMPUTED in O(n) TIME [LINGAS '94] STARTING FROM DT AND DELETING.

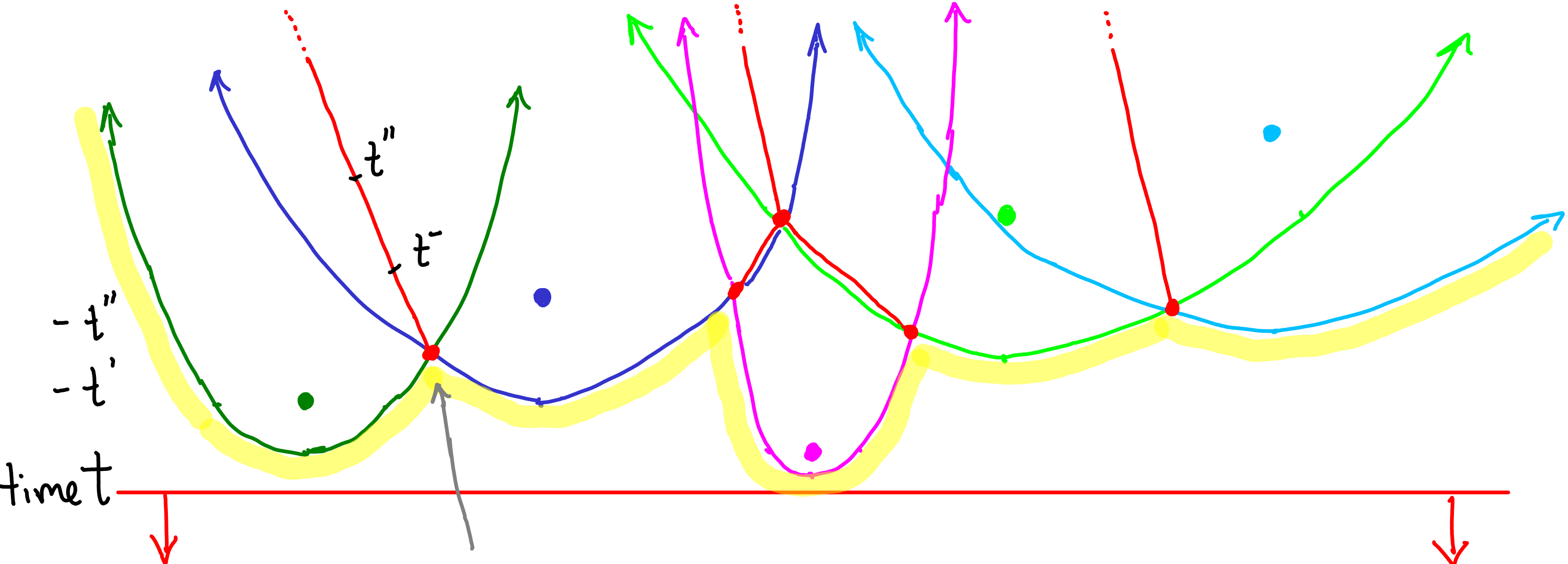
FORTUNE'S LINE-SWEEP ALGORITHM FOR VORONOI DIAGRAMS

- REGULAR LINE-SWEEP CANNOT WORK

  IN THE SENSE THAT EVERYTHING "BEHIND" THE LINE IS FINAL.

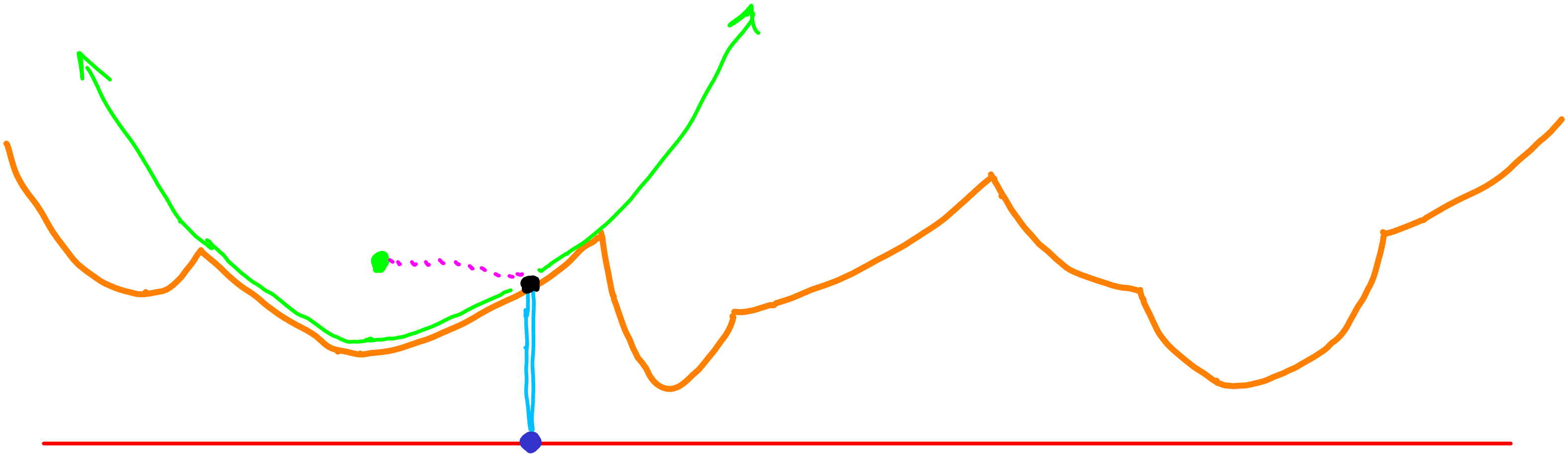


EVERYTHING (above) BEHIND THE BEACHFRONT IS FINAL.



- EVERY NODE ON THE BEACHFRONT REPRESENTS THE PROPAGATION OF A VORONOI EDGE.

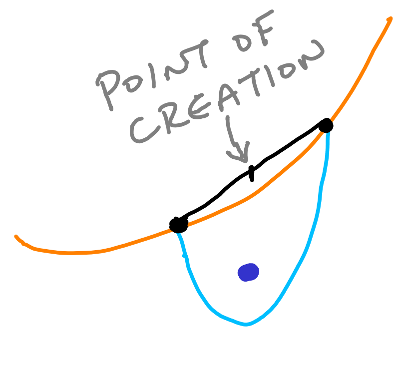
- PARABOLAS CHANGE SHAPE AS SWEEPLINE CONTINUES.



- WHEN PASSES THROUGH A NEW POINT, CREATE A PARABOLA.
- IT INTERSECTS THE BEACHFRONT AT AN EXISTING PARABOLA.

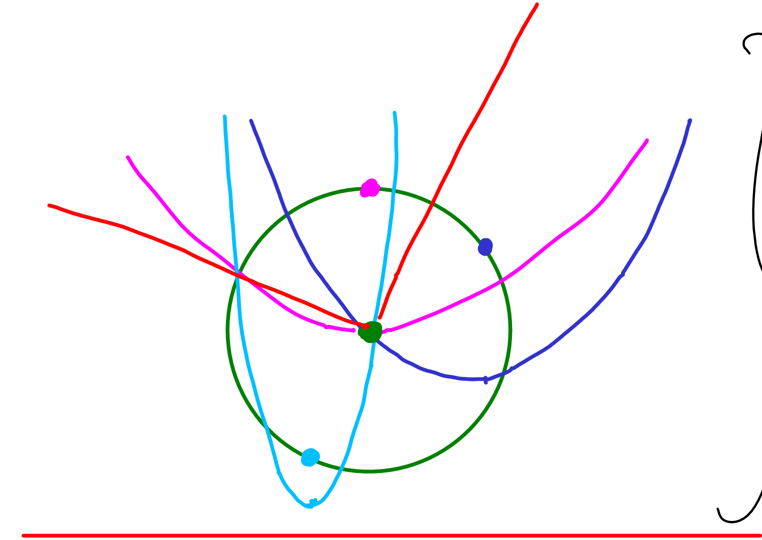
  $d(\bullet\bullet) = d(\bullet\bullet) \Rightarrow \bullet$ is ON VORONOI EDGE.

- AS CONTINUES, $\bullet$ GROWS LEFT & RIGHT, FOLLOWING INTERSECTION $\cap$



THIS IS THE ONLY WAY TO ADD TO - SEE TEXTBOOK.

↳ BEACHFRONT HAS SIZE $O(n)$



CREATION OF VORONOI VERTEX:

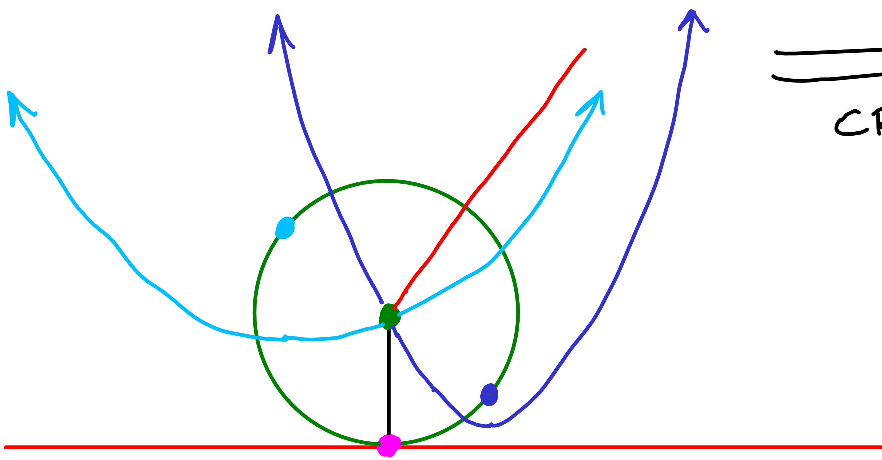
U & U MEET AND ELIMINATE U
→ ←

3 PARABOLAS INTERSECT at •

U U U : EQUAL DIST. TO /

THUS • • • ARE ON ○

NOW THAT U IS GONE, U & U CONTINUE FORMATION OF VORONOI EDGE (MERGED)



CREATION TYPE II