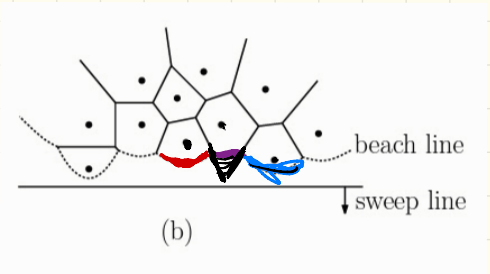
Fortune's Algorithm

3/18

Sweep line Status:
- y-coord of the sweepline
- left-to-right order of sites that define beach line

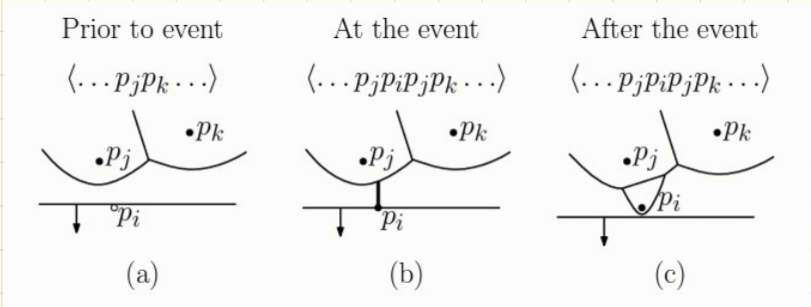
DON'T STORE THE ARCS



Events
- site event: sweepline passes over new site → new arc added to beachline
- vertex event: length of arc goes to 0 → a new Voronoi vertex

*Site Event
- sweepline passes site p_i
- shoot a ray up to identify parabola above p_i
- insert p_i as rep into beachline

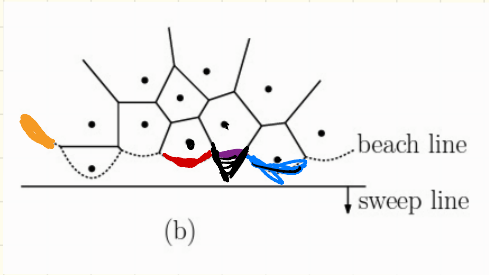
How many new arcs can be introduced
create 1 new arc and split an old arc into 2
⇒ at most $2n-1$ arcs on the sweepline



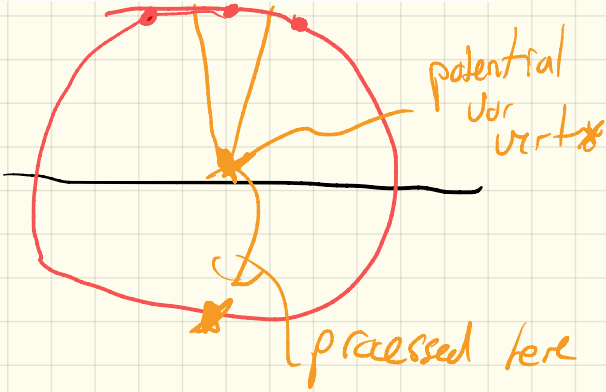
Vertex events:

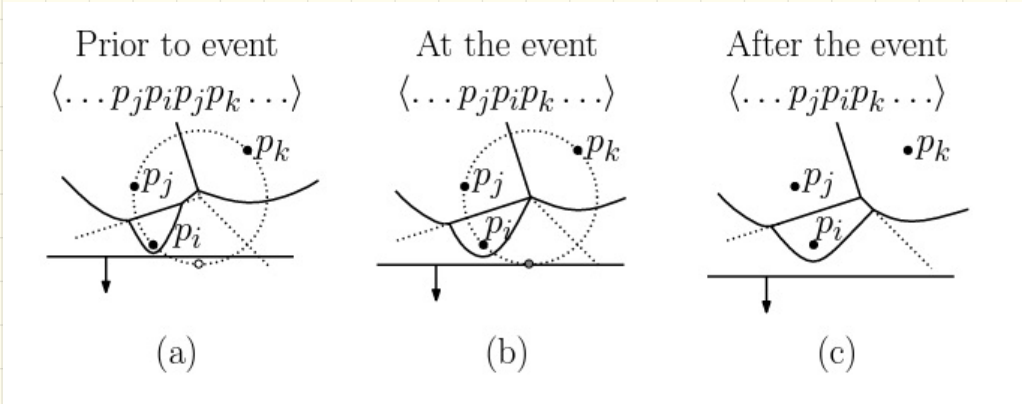
- generated dynamically as triples of sites that are adj on the beachline
- event: a circumcircle of 3 points that define potential Voronoi vertex

set event time by lowest point in the circle



(b)

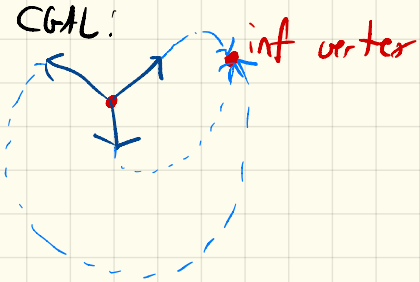




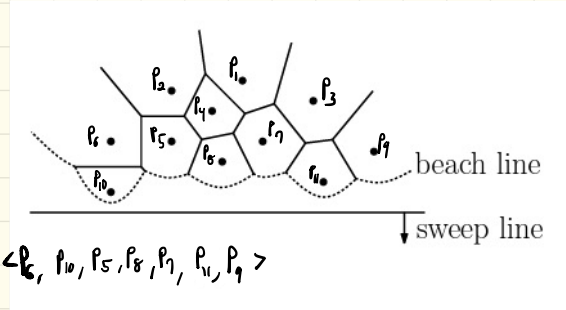
Sweepline algo:

Partial Voro. diagram: Vor dgm computed "so far"
stored in a DCEL (or any other planar subdivision D.S.)

CGAL:



Beach line:
- sorted seq of sites (sorted by arcs)



$\langle p_6, p_{10}, p_5, p_8, p_7, p_{11}, p_9 \rangle$

- stored in <u>dictionary</u>
- key op: "find arc above a site"
  How to find?
  - Binary search on breakpoints
  - two sites in the beachline p_i, p_j
    breakpoint for p_i, p_j at y-coord of sweep line
    $\|p_i - q\|^2 = \|p_j - q\|^2$ w/ $q \in \mathbb{R}^2$
    expand out and solve for q_x
    plug in y-coord

Ops:

Search: given y-coord of sweepline
and a site p_i
determine arc above p_i
return a ref to the beach line entry

Insert and split: insert new entry w/in arc p_j

$\langle \cdots, p_i, \cdots \rangle \rightarrow \langle \cdots, p_j, p_i, p_j, \cdots \rangle$

Delete given a ref to p_j on the beach line dele entry

$\langle \cdots, p_i, p_j, p_k, \cdots \rangle \rightarrow \langle \cdots, p_i, p_k, \cdots \rangle$

Time:

w/ slight modification $O(\log n)$ per op as D.S. cont have more than $2n$ objs

Event queue
- insert and delete events
- extract event w/ largest y

Time:

$\log(k)$ for P.Q w/ max size k (proportional to)

Create vertex events
- each consecutive tripple on beach line adds to event que

$\cdots, p_i, p_j, p_k, \cdots$

compute circumcircle

if lowest point on the circle is below the sweepline
$\Rightarrow$ add to event queue
store a link to the sites that generate it

Alg:

Init:

insert all site events

pop event (... site event), let p_i be the site

insert p_i into sweepline status (init beachline)

Until event queue is empty:

pop event

handle event

Site event

- advance sweepline to p_i
- search on beachline for arc above p_i (let it be p_j)
- apply insert-split op to insert p_i into the beachline at p_j
- create a (dangling) edge for the Voronoi diagram at the bisector of p_i, p_j
- remove events w/ old triples that are no longer w/ p_j
- add new events of triples w/ p_i (no event for $p_j p_i p_j$)

Vertex event: let p_i, p_j, p_k be sites that generate the event (order L→R)

- delete entry for p_j from the beachline
- create a Voronoi vertex in dgm (at the circumcenter of p_i, p_j, p_k) call it v
- join edges to v (p_i, p_j) and (p_j, p_k)
- create new (dangling) edge for p_i, p_k bisector
- delete any vertex events w/ trips w/ p_j
- generate new events for triples w/ p_i & p_k (at most 2)

Analys:
each event:
$O(1)$ process the event type;
$O(1)$ ops on the data structures
size of all the data structures are $O(n)$
$\Rightarrow$ all ops are $O(\lg n)$
$\Rightarrow$ total time is $O(n \lg n)$
and space is $O(n)$!!