CPSC 420 Lecture 4 : Today's announcements:

- HW1 available on Gradescope, due Jan 19, 23:59
- Examlet 1 on Jan 27 in class.
- Reading: Voronoi Diagrams [Computational Geometry: Algorithms and Applications 3rd Edition pg 147]
- Reading: Linear Programming [Intro to Algs 4th Ed. by Cormen, Leiserson, Rivest, Stein Ch. 29-29.2, pg 817]

Today's Plan

- Chan's Algorithm - analysis
- Voronoi diagrams

Chan's Algorithm with guess g for CH size

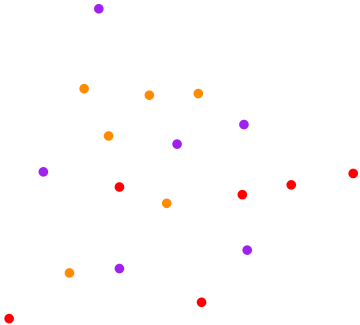
Given n points P and a guess g for the number of hull points...

1. Divide P into n/g groups of g points
2. Use Graham's Scan to find the convex hull of each group in $O(g \log g)$ time per group

Chan's Algorithm with guess g for CH size

Given n points P and a guess g for the number of hull points...

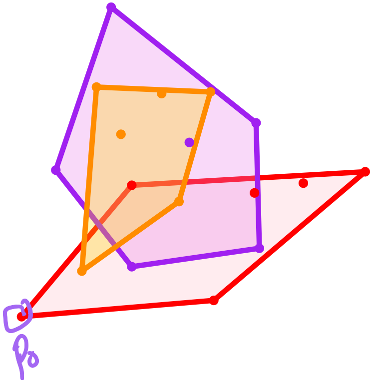
1. Divide P into n/g groups of g points
2. Use Graham's Scan to find the convex hull of each group in $O(g \log g)$ time per group



Chan's Algorithm with guess g for CH size

Given n points P and a guess g for the number of hull points...

1. Divide P into n/g groups of g points
2. Use Graham's Scan to find the convex hull of each group in $O(g \log g)$ time per group



Chan's Algorithm with guess g for CH size

Given n points P and a guess g for the number of hull points...

$O(n)$ 1. Divide P into n/g groups of g points

$O(n \log g)$ 2. Use Graham's Scan to find the convex hull of each group in $O(g \log g)$ time per group

$O(n)$ 3. Find the lowest point p_0

$i=0$

$O(n \log g)$ 4. Gift-wrap (Jarvis March) these convex hulls for g wrap steps. To find the next hull point p_{i+1}

   4.1 find the right-tangent from p_i to each group hull in $O(\log g)$ time per group — $O(n/g \log g)$

   4.2 p_{i+1} is rightmost-by-tangent-angle of these tangent points — $O(n/g)$

   4.3 If $p_{i+1} = p_0$ output hull Succeed

5. Output "g is too small!" Fail

Total time: $O(n \log g)$.

How to generate guesses

Chan's Main CH Algorithm

Run Chan's Alg. with guess $g = 4$ then $g = 16$ then $g = 256$ ...
until it outputs the Convex hull

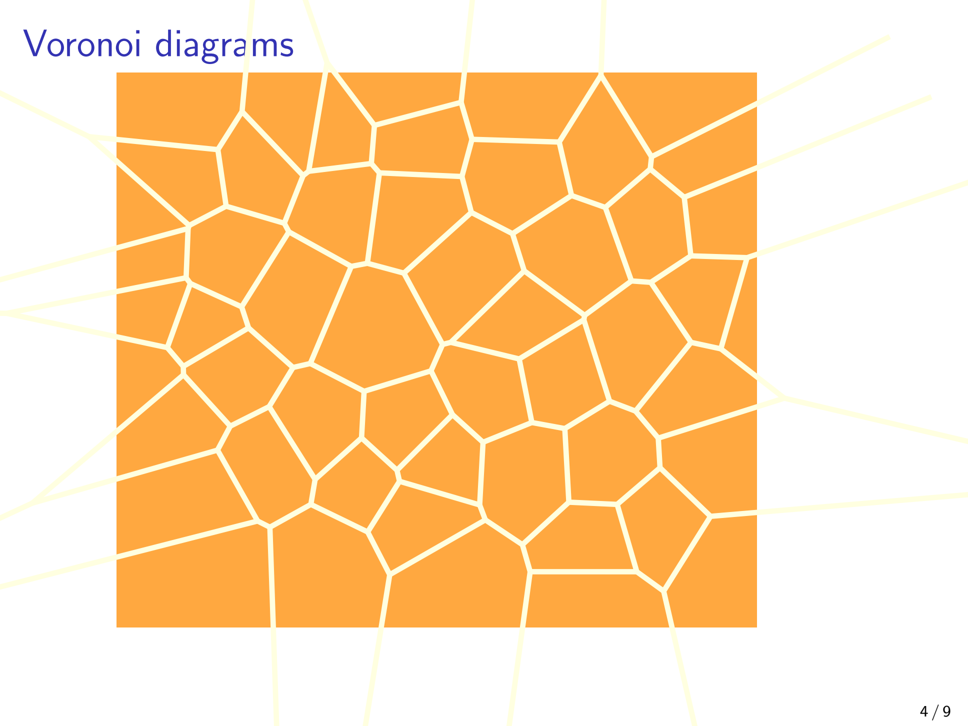
$$g = 2^{2^t} \text{ on the } t^{\text{th}} \text{ try.}$$

$$2^{\lceil \lg \lg h \rceil + 1} \le 4 \lg h$$

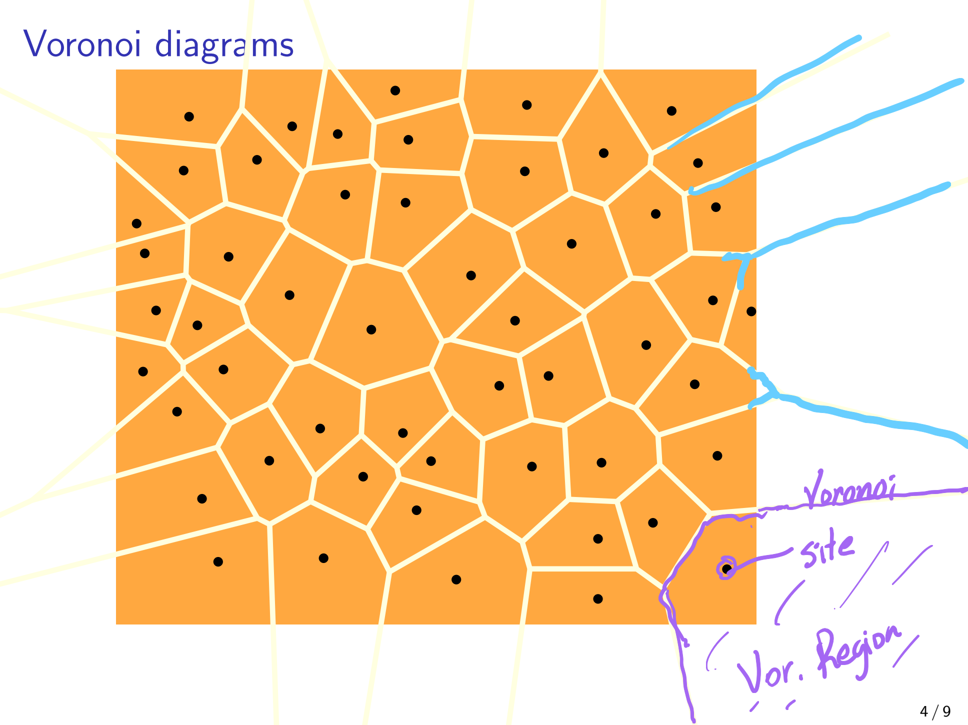
Total run time (until $g \geq$ hull size h):

$$\sum_{t=1}^{\lceil \lg \lg h \rceil} O(n \log(\underbrace{2^{2^t}}_{g})) = \sum_{t=1}^{\lceil \lg \lg h \rceil = x} O(n 2^t) = O(n \sum_{t=1}^{\lceil \lg \lg h \rceil} 2^t) = O(n \lg h)$$

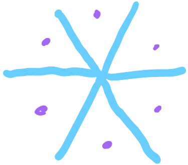
Voronoi diagrams



Voronoi diagrams



Voronoi diagrams



Definition

A **Voronoi diagram** *of a set of n sites (points)* $s_1, \ldots, s_n$ *is a set of regions* $R_1, \ldots, R_n$ *where* R_i *is the set of points* x *such that* $d(x, s_i) \leq d(x, s_j)$ *for all* j.

A **Voronoi edge** *is the border between two regions:* $\{x | x \in R_i \text{ and } x \in R_j \text{ and } i \neq j\}$.

A **Voronoi vertex** *is the intersection of Voronoi edges:* $\{x | x \text{ in more than 2 Vor. Regions}\}$.

Problem Given $S = s_1, s_2, \ldots, s_n$, find Voronoi vertices and edges

Voronoi Diagram using Divide and Conquer [Shamos & Hoey '75]

L = set of $n/2$ to left of R

R = set of $n/2$ to right of L

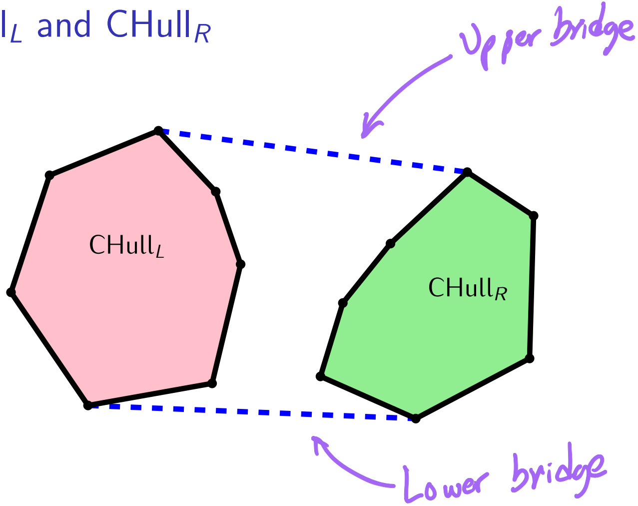
This algorithm finds both VorD and CH of S.

0. Sort sites by x-coord

VorD+CHull(S)

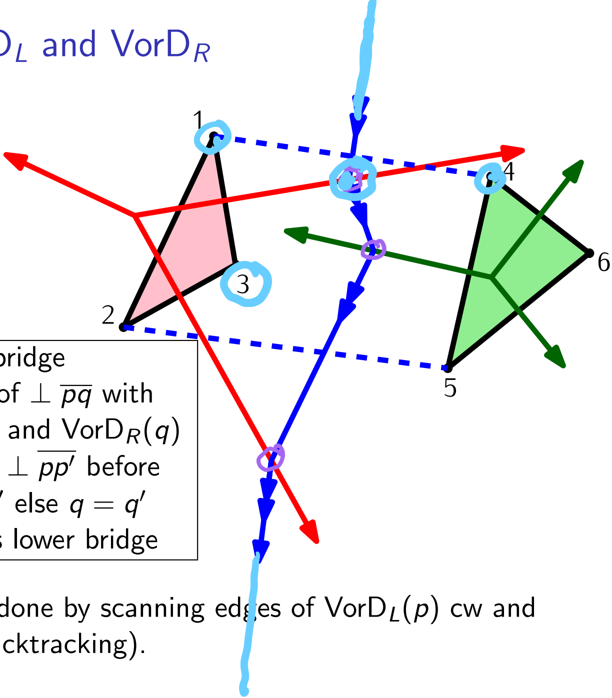
1. if $|S| = 1$ return VorD $= \emptyset$ and CHull $= [s_1]$
2. if $|S| = 2$ return VorD $= \perp$ -bisector of $\overline{s_1 s_2}$ and CHull $= [s_1 s_2]$
3. Recursively find VorD$_L$ and CHull$_L$ of $s_1, \ldots, s_{n/2}$
4. Recursively find VorD$_R$ and CHull$_R$ of $s_{n/2+1}, \ldots, s_n$
5. Merge CHull$_L$ and CHull$_R$ to get CHull
6. Stitch together VorD$_L$ and VorD$_R$ to get VorD
7. Return VorD and CHull

Merge CHull_L and CHull_R



See your Homework 1 for a (more general) $O(n)$ time solution.
The two blue lines are the upper and lower bridges.

Stitch together VorD_L and VorD_R



1. Let $\overline{pq}$ be upper bridge
2. Find intersection of $\perp \overline{pq}$ with edge of $\text{VorD}_L(p)$ and $\text{VorD}_R(q)$
3. If $\perp \overline{pq}$ intersects $\perp \overline{pp'}$ before $\perp \overline{qq'}$ then $p = p'$ else $q = q'$
4. Repeat until $\overline{pq}$ is lower bridge

Note: Step 2 can be done by scanning edges of $\text{VorD}_L(p)$ cw and $\text{VorD}_R(q)$ ccw (no backtracking).

Example from Computational Geometry by Preparate & Shamos