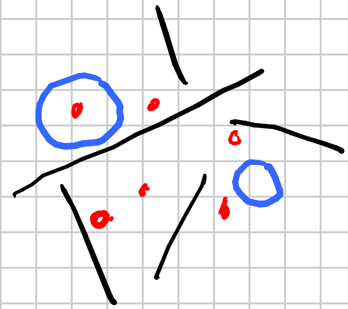
Computational Geometry

Notetitel 22-08-2008

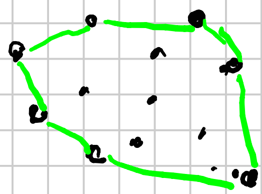
Objects (in 2D, 3D, ...)

- Points
- Lines (Halfspaces)
- Circles (Balls)

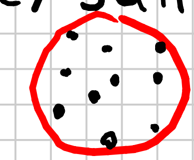


Problems

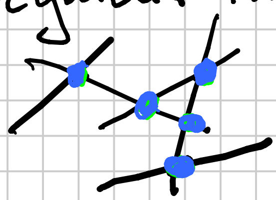
Convex Hull



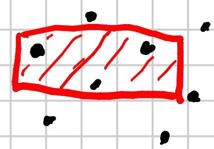
Smallest enclosing circle/ball



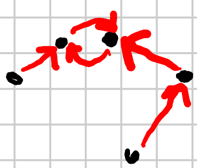
Segment intersection



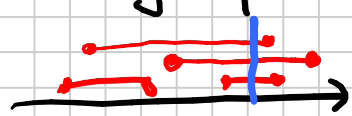
Orthogonal range searching



Nearest Neighbors

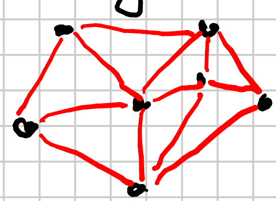


Stabbing queries

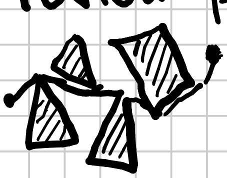


Measure of rectangles

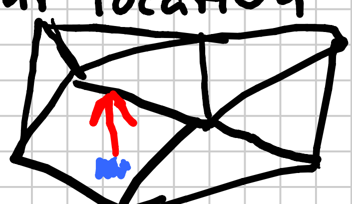
Triangulations



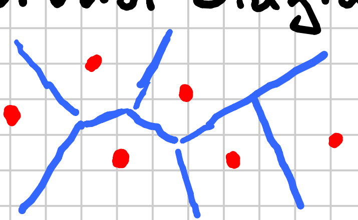
Motion planning



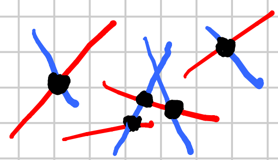
Point location



Voronoi Diagram



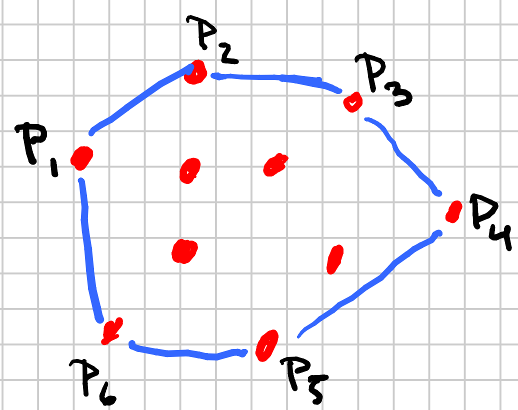
Red-Blue intersection



Course

- Geometric objects (Mathematics)
- Algorithms and Data Structures (Theory)
- Implementations (Practice, Projects)

Convex Hull (2D)

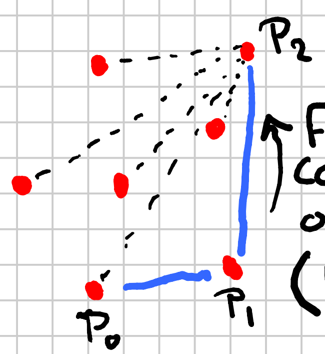


Input: n points

Output: Smallest polygon containing points (size h)

Observation: CH is defined by a subset of the points

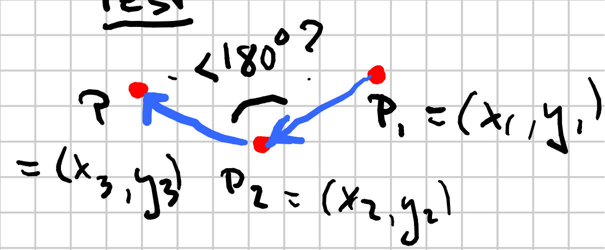
Algorithm 1 : Gift wrapping $O(n \cdot h)$



special case 3 points on a line ?

Find points in counter clockwise order (brute force)

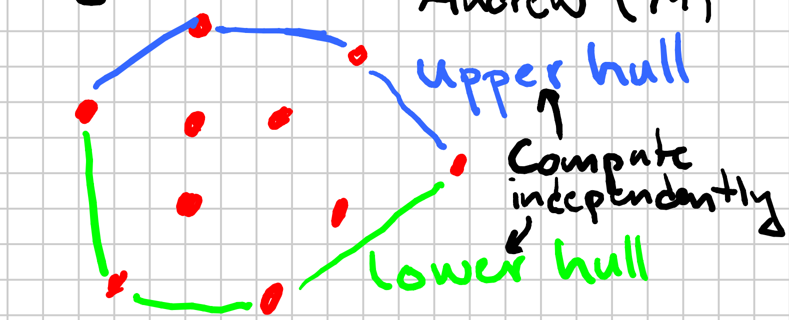
Test

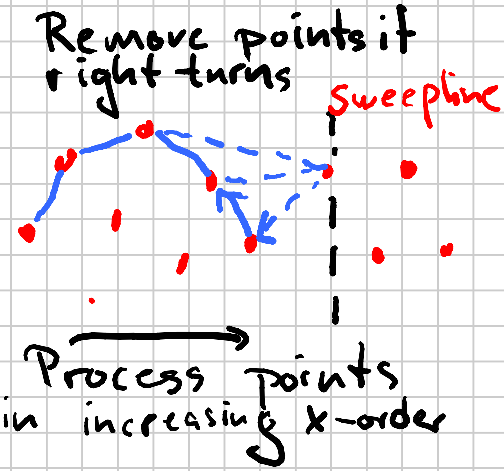


$$\Delta = \begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} = x_1(y_2 - y_3) - x_2(y_1 - y_3) + x_3(y_1 - y_2)$$

- $\Delta < 0$: Right turn
- $\Delta = 0$: On a line
- $\Delta > 0$: Left turn

Algorithm 2: Graham's scan (72) $O(\mathrm{Sort}(n) + n)$
Andrew (79)





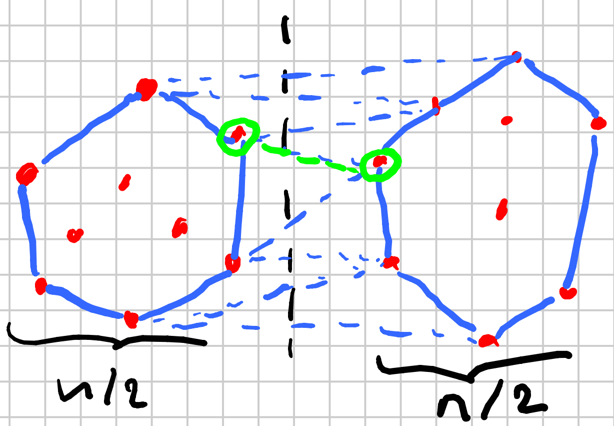
Analysis:

- Sort points w.r.t. x-axis — $O(\mathrm{Sort}(n))$
- Insert each point once — $O(n)$
- Delete each point at most once — $O(n)$

$\overline{\quad O(\mathrm{Sort}(n) + n) \quad}$

Note: If points sorted w.r.t. x $\Rightarrow$ $O(n)$ time.

Algorithm 3: Divide-and-conquer $O(n \cdot \log n)$



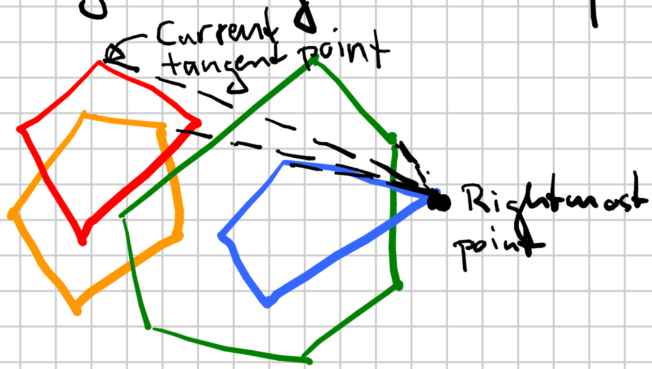
- Divide wrt x-median – selection, $O(n)$ time
- Solve recursively left-and-right
- Connect extreme points
- Repeatedly remove concave points

$$T(n) = 2 * T\left(\tfrac{n}{2}\right) + O(n) \Rightarrow T(n) = O(n \cdot \log n)$$

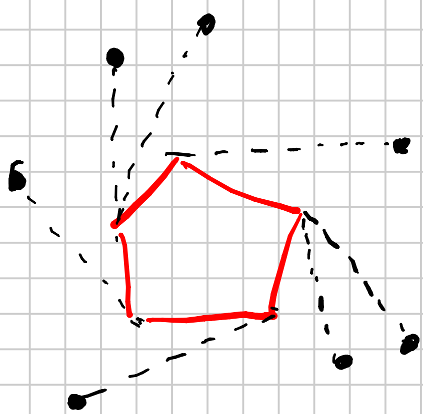
Algorithm 4: Kirkpatrick & Seidel (86), Chan (96)

– Output sensitive $O(n \cdot \log h)$

- m a parameter ($m = 2^{2^i}$)
- Partition input into $\lceil n/m \rceil$ groups each of size $\leq m$ $O(n)$
- Construct CH for each group $O(\frac{n}{m} m \cdot \log m)$
- Perform giftwrapping for $\leq h$ steps starting at rightmost point



Trick Start next search at a group where the previous one ended.



The tangents touching a group during gift-wrapping.

$$O(h \cdot \frac{n}{m} + n)$$

#wrappings

#groups

Total time to scan around the groups

Total time $O(n \cdot \log m + \frac{h}{m} \cdot n)$

Trick Stop wrapping after m steps $O(n \cdot \log m)$

Trick Choose $m = 2^{2^i}$ for $i = 0, 1, 2, \ldots$ until $m \geq h$

(i.e. $m < h^2$)

$$O\left(\sum_{i=0}^{\log\log h - 1} n \cdot \log 2^{2^i}\right) = O\left(n \cdot \sum_{i=0}^{\log\log h+1} 2^i\right) = O(n \cdot \log h)$$