September 10, 2024 2:22 PM

Convex Hull in 2D

Alg'm 0: brute-force $\Rightarrow O(n^3)$ time

Alg'm 1: Jarvis' march $\Rightarrow O(n^2)$ time [similar to selectionsort]

Alg'm 2: Graham's scan $\Rightarrow O(n \log n)$ time [similar to insertionsort & sweeping]

Alg'm 3: "mergehull" (Preparata-Hong) $\Rightarrow O(n \log n)$ time

[Similar to mergesort]

lower bd $\Omega(n \log n)$ in worst case

what if the # of UH vertices (the "output size") is small?

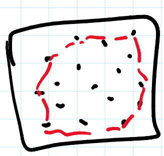
h

e.g. Graham's scan: $O(n \log n)$

Jarvis march: $O(nh)$

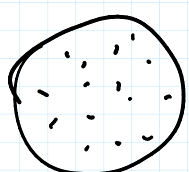
Known: for pts unif. distributed inside square,

$$E[h] = \Theta(\log n).$$



for pts unif. distributed inside circle,

$$E[h] = \Theta(n^{1/3})$$



$\vdots$

what's best worst-case runtime in n & h?

("output-sensitive alg'ms")

Answer: $O(n \log h)$ Kirkpatrick-Seidel '86

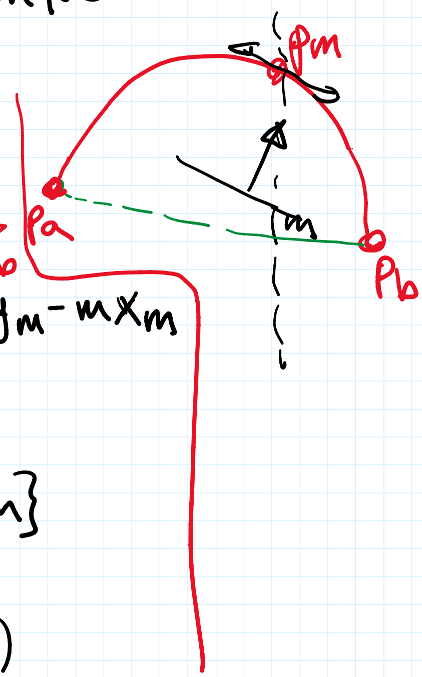
$O(n \log h)$ C.-Snoeyink-Yap '95 (simplification)

$O(n \log h)$ — C.-Snoeyink-Yap '95 (simplification)

$O(n \log h)$ — C. '95

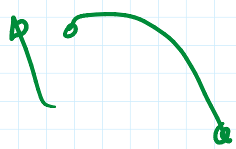
Alg'm 4: Quickhull ('77)

idea - quicksort-style divide & conquer



quickhull(P, p_a, p_b):

0. remove pts below $\overleftrightarrow{p_a p_b}$ from P
1. pick a slope m, e.g. slope of $\overleftrightarrow{p_a p_b}$
2. find $p_m \in P$ maximizing $y_m - m x_m$ — $O(n)$ time
   $= (x_m, y_m)$
3. $P_L = \{ p_i \in P : x_i < x_m \}$
4. $P_R = \{ p_i \in P : x_i > x_m \}$
5. quickhull(P_L, p_a, p_m)
6. quickhull(P_R, p_m, p_b)



Analysis: $T(n,h) = T(n_1,h_1) + T(n_2,h_2) + O(n)$

$h = \#$ hull edges

where $n_1 + n_2 \le n$ (+1)
$h_1 + h_2 = h$ (+1)

$T(n,1) = O(n)$

$\Rightarrow T(n,h) = \boxed{O(nh)}$ worst-case

when division is unbalanced.

Alg'm 5: C.-Snoeyink-Yap '95

idea. modify quickhull ...
with clever choice of pivot m
& more pruning

with clever choice of pivot ...
& more pruning

how to pick slope m (line 1)?

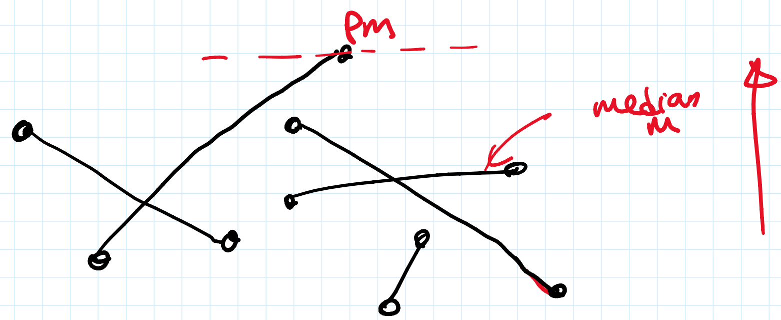
1! pair up points & pick median slope

i.e. write $P = \{p_1, p_2, p_3, \ldots, p_n\}$

$O(n)$ time

form $n/2$ pairs $p_1 p_2$, $p_3 p_4$, ...

m = median of slopes of $\overleftrightarrow{p_1 p_2}$, $\overleftrightarrow{p_3 p_4}$, ...



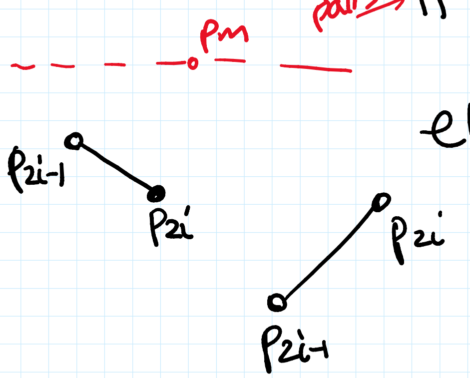
add pruning steps before lines 5-6:

say p_{2i-1} is left of p_{2i}

n/4 pairs → if $\overleftrightarrow{p_{2i-1} p_{2i}}$ has slope $< m$,

remove p_{2i} from P_L (if it was in P_L)

else remove p_{2i-1} from P_R (if it was in P_R)



Analysis: after pruning step,

$|P_L| \leq \frac{3}{4} n$

$|P_R| \leq \frac{3}{4} n$

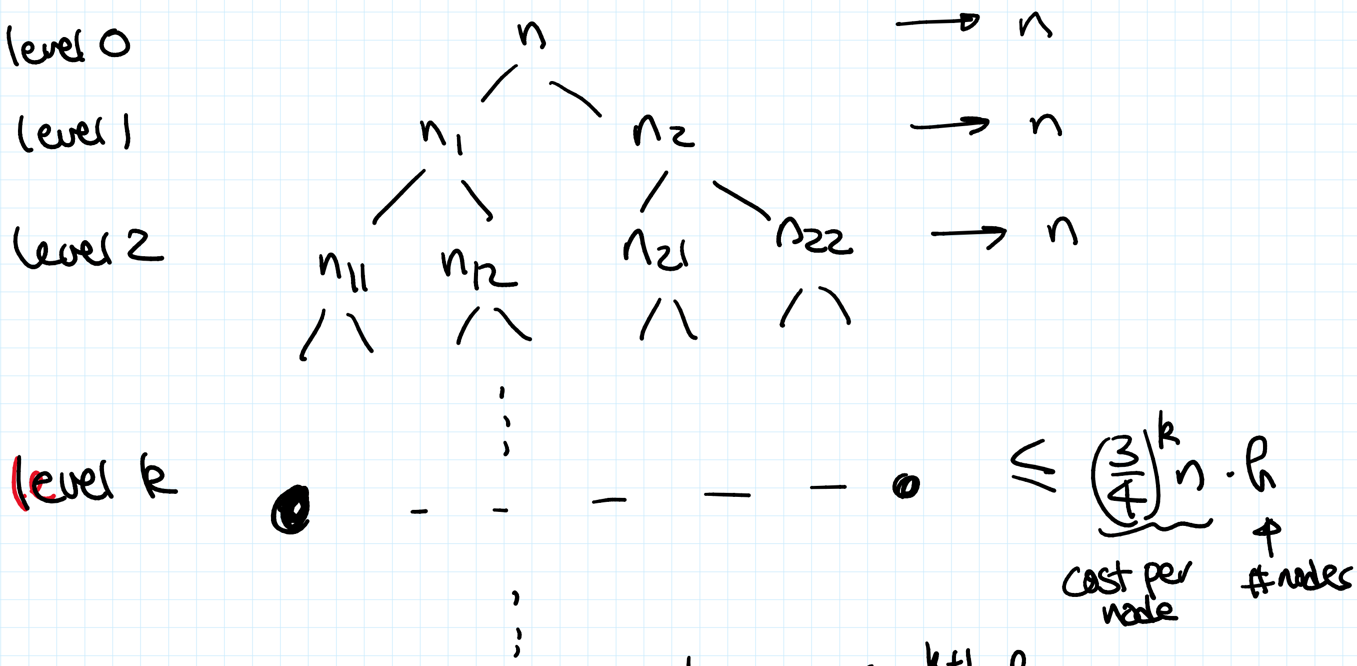
$T(n, h) = T(n_1, h_1) + T(n_2, h_2) + O(n)$

$n_1 + n_2 \leq n$ (+1)

$\underline{h_1 + h_2 \leq h}$ (+1)

$\boxed{n_1, n_2 \leq \frac{3}{4} n}$

solve by recursion tree



$$\text{Total:} \quad nk + \left(\tfrac{3}{4}\right)^k nh + \left(\tfrac{3}{4}\right)^{k+1} nh + \cdots$$

$$= O\left(nk + \left(\tfrac{3}{4}\right)^k nh\right)$$

set $k = \log_{4/3} h$

$$= O(n \log h + n)$$

$$= \boxed{O(n \log h)}$$

Alg'm 6: (C. '95)

idea - combine Graham & Jarvis

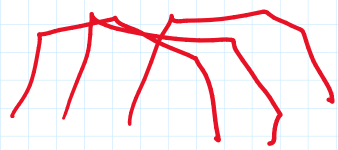
- by "grouping"

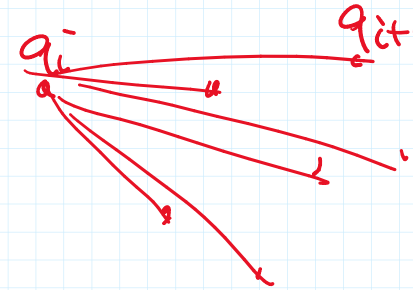
divide P into groups P_1 P_2 -- $P_{n/h}$

1. arbitrarily divide P into groups $P_1, P_2, \ldots, P_{n/h}$ each with h pts

2. for $k = 1$ to n/h do
   U_k = UH of P_k by Graham scan

$O(\frac{n}{h} \cdot h \log h)$
$= O(n \log h)$

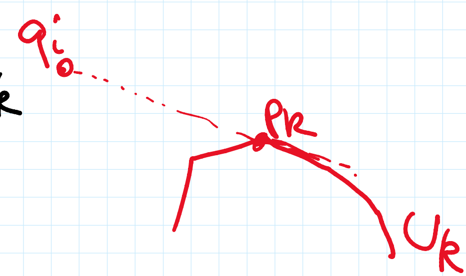




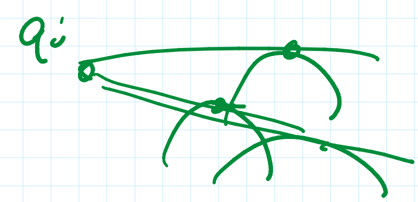
3. q_1 = leftmost pt
4. for $i = 1$ to h do {
5. if q_i = rightmost pt return $\langle q_1, \ldots, q_i \rangle$
6. q_{i+1} = any pt right of q_i
7. for $k = 1$ to n/h do {

$O(\log h)$

p_k = tangent pt between q_i and U_k



if p_k above $\overleftrightarrow{q_i q_{i+1}}$
   $q_{i+1} = p_k$



}

$O(\frac{n}{h} \log h)$

$O(h \cdot \frac{n}{h} \log h) = O(n \log h)$

Total: $\boxed{O(n \log h)}$