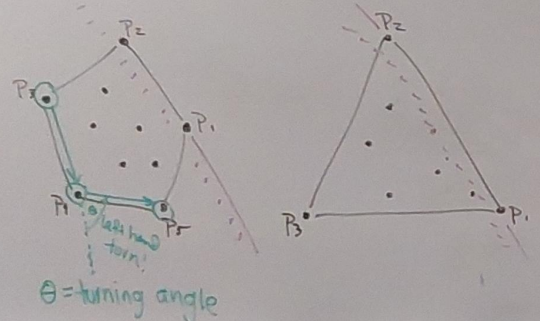
(Planar)

Convex Hull

INPUT: $P \subset \mathbb{R}^2$, $|P| < \infty$

OUTPUT: a list of verts on the convex hull, in CCW order.

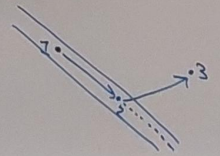


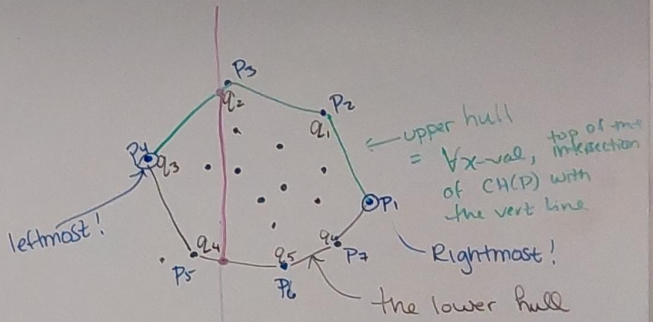
Lemma 1:

① verts def. CH are in P

② for any edge $e \in \partial CH(P)$ all points in P are on the same side of it.

③ three consec. verts on $\partial CH(P)$, in ccw order form a left hand turn





$CH(P) = \langle p_1, p_2, p_3, p_4, p_5, p_6, p_7 \rangle$

$= \langle q_1, q_2, q_3, q_4, q_5, q_6, q_7 \rangle$

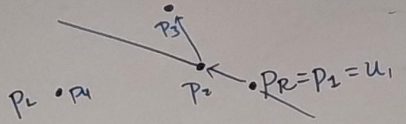
$3 \leq$ # points in $CH \leq n = |P|$ in $\mathbb{R}^2$

↑ assuming GP, $|P| \geq 3$

Thought experiment:

$CH\left(\begin{array}{c}\cdot\\ \cdot\\ \cdot\\ \cdot\\ \cdot\end{array}\right) = a \text{ —— } d$

colinear in $\mathbb{R}^2$



$P = \langle p_1, p_2, \ldots, p_n \rangle$, sorted R to L

output $U = \langle \underbrace{\overset{p_R}{u_1}, u_2, \ldots, \overset{p_L}{u_n}}_{\text{upper hull}} \rangle$, sorted CCW (lower hull)

Idea / thoughts:

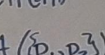
- I know leftmost (p_L) / rightmost (p_R)
- want to find UCH, from p_R to p_L, CCW
  this is also decreasing x-value order.
- iterative algo: find UCH for largest x-value points, add one more point + fix it.

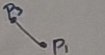
Cool algo to know:

I can find the k^{th} largest element in $\Theta(n)$ time!
• max • min • median.

Running CH:

Add p_1 $\rightsquigarrow$ UCH($\{p_1\}$) = p_1

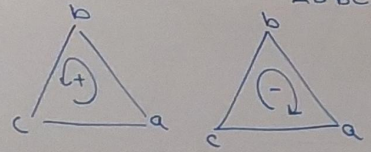
Add p_2 $\rightsquigarrow$ UCH($\{p_1, p_2\}$) = 

Add p_3 $\rightsquigarrow$ UCH $\{p_1, p_2, p_3\}$ = 

Recall: Jack's Favorite Determinant:

$$\text{orient}(a,b,c) = \text{sign}\left(\begin{vmatrix} 1 & a_x & a_y \\ 1 & b_x & b_y \\ 1 & c_x & c_y \end{vmatrix}\right)$$

this value is twice the signed volume of the triangle $\triangle abc$.



Graham's Scan (P)

1: $\langle p_1, p_2, \ldots, p_n \rangle \leftarrow$ sort (P) $\quad \Theta(n \log n)$
(rightmost, leftmost; sorting on x-coordinate)

2: $S \leftarrow$ a stack init by pushing p_1, then p_2 $\quad \Theta(1)$

3: $i = 3$ (my current rep. of the CH. Needs p_1 in.)

4: while $i \le n$

5: if orient$(S(t-1), S(t), p_i) \le 0$ and $|S| \ge 2$ $\quad \Theta(n)$

6: S.pop()

7: endif

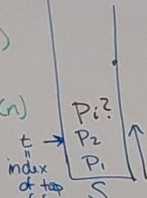
8: S.push(p_i) ← Since p_i is left most of $P_i = \{p_1, p_2, \ldots, p_i\}$, we know $p_i \in \partial CH(P_i)$

9: i++

10: end for $\quad \Theta(n)$

11: return S as an array (top of S = end of array)

$\Theta(n \log n + 1 + n + n) = \Theta(n \log n)$



reading this way, I visit verts of tentative UCH in CCW order.

if  c to right of ab orient(a,b,c)<0

then  b to left of ac.

orient(a,c,b) > 0

Partial Correctness: loop invariants!

"the LI builds up the solution"

LI: S (bot to top) is the UCH of $\{P_1, P_2, \ldots, P_{in}\}$

Must prove:

1) INITIALIZATION $(P \Rightarrow L)$
   "true going into the loop"

* 2) MAINTENANCE $(L_i \wedge G \Rightarrow L_{i+1})$
   "if true going into ith iteration, still true going out of it. Might break in between"

3) END: $L_i \wedge \neg G \Rightarrow Q$
   "if the loop ends, it was correct" and c_i holds

much like induction!

Statements

L = loop invariant
(L_i = loop inv. c_i attempting to go into ith loop)
↳ = coming out of i-1 iteration

P = preconditions (what is true before entering the loop)

G = loop guard

Q = post-condition
↖ state this first!

LI: shows If the algo terminates, then it was correct.

So, we also need to show it terminates!
(often, give the RT)

RT: # pop + # push = # orientation tests

on ith iteration:
1 push
C_i = # of pops

in total over whole while loop

$$\sum_{i=3}^{n} C_i + 1 = \sum_{i=3}^{n} C_i + \sum_{i=3}^{n} 1 = \#\text{pops} < \Theta(n)$$

$$\sum C_i \leq \#\text{pops} = O(n)$$