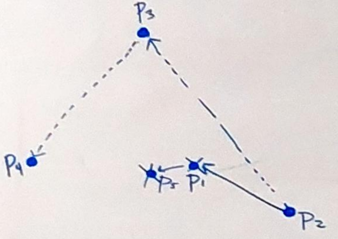


Graham's Scan for Convex Hull

Worst-case runtime: $\Theta(n\log n)$

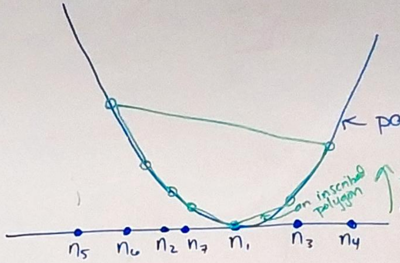
also the worst-case RT of comparison based sort using real-RAM MOC

Q: Is this the best I can do?

Prob A (Sorting) $\xrightarrow{\text{reduce it to}}$ Prob B (convex hull)

take input of this problem $\xrightarrow[\text{poly time + reversible}]{\text{change it}}$ to input of other prob

ex: traveling salesman $\rightsquigarrow$ New Prob

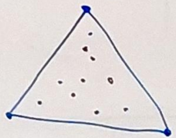


reduction: $n_i \to (n_i, n_i^2)$

INPUT: #s I want to sort

inscribed polygon: set of points in the green curve, connected in order.

Worst case: can't really do better, but...

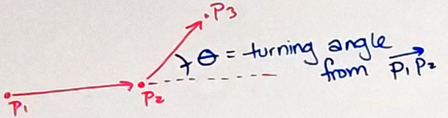


n = # of points

h = # of points on hull (the size of the output)

Output-sensitive CH Algo 1:

Jarvis March / Giftwrapping



runtime will be: $\Theta(nh)$

- If $h = \Theta(n)$, we're worse-off $\Theta(n^2)$ Graham wins!
- If $h = \Theta(1)$, this is great!! $\Theta(n)$ Jarvis wins!
- If $h = \frac{1}{2}n = \Theta(n)$
- If $h = \Theta(\log n)$, we flip a coin. Either way, $\Theta(n \log n)$

Want: $h = o(\log n)$

JarvisCH(P)

$\Theta(n)$ — $v_1 = (v_1^{(1)}, v_1^{(2)}) \leftarrow$ point w/ lowest y-value

$\Theta(1)$ $\left\{ \begin{array}{l} v_0 = (-\infty, v_1^{(2)}) \\ i = 1 \end{array} \right.$

while $v_i \neq v_1$ and not first time through

    $v_{i+1} \leftarrow$ point p in $P \setminus \{v_1, \dots, v_i\}$ that min turn angle from $\overrightarrow{v_{i-1} v_i}$

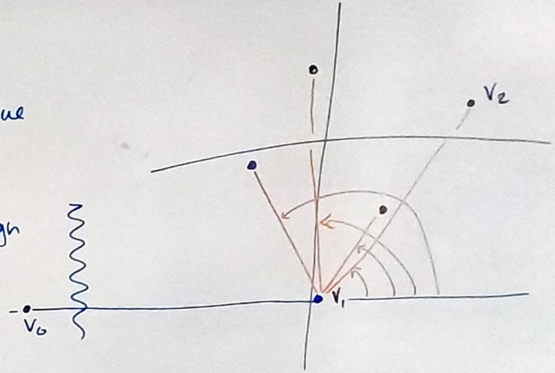
    i++

end while

h times, check $n-i$ pts each time: $\sum_{i=1}^{h} n - i = \Theta(nh)$

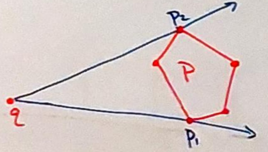
$\Theta(h)$ — return $\langle v_1, \dots, v_h \rangle$

$\Theta(n + 1 + nh + h) = \Theta(nh)$



Chan's Algorithm

- RT: $\Theta(n \log h)$
  $\Rightarrow$ can't sort anything too large!
- another tool
  given polygon P, pt q outside P
  ↳ def. by verts in CCW order



find p_1 and $p_2 \in P$ such that P is "inside" the cone defined by the rays $\overrightarrow{qp_1}$ and $\overrightarrow{qp_2}$

RT: $\Theta(\log n)$ → exercise: find algo.

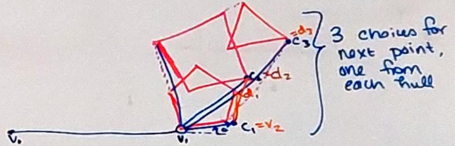
1st assume h is known.

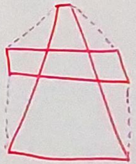
Chan's Algo has 2 main steps:

① Divide: Compute mini-hulls using Graham

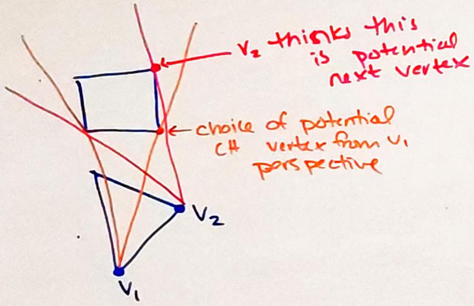
1. $k \leftarrow \lceil n/h \rceil$
2. partition P into k equal sets, $P_1, P_2, \ldots, P_k$.
   note: $|P_i| = h \pm 1 = \Theta(h)$
3. Use Graham to find CH of each P_i
   result: $H_1, H_2, \ldots, H_k$

② Merge: each "mini-hull" as a fat point & use Jarvis on the mini hulls.





the choices change as we add pts to CH.



If: h too large, say $h = \Theta(n\log n)$, then step 1 is too long.

Sol'n: Ensure h doesn't get too big.

If h^* optimal, $h < h^* < h^2$

If: h is too small

Sol'n: once add too many pts on hull, restart.