DIVIDE AND CONQUER ALGORITHM

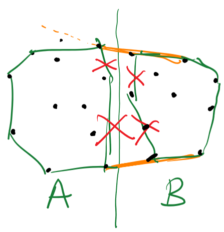
(recursive algorithm — alg. that calls itself)

Pseudocode:

Sort the points by x-coordinate

let: A = left half of the points

B = right half of the points



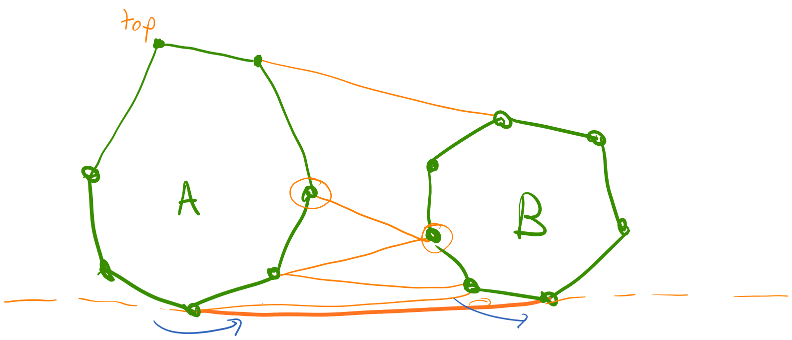
Find the convex hulls of A and B — use the divide-and-conquer alg.

- If I get down to 3 points or less, then return those points in CCW order.

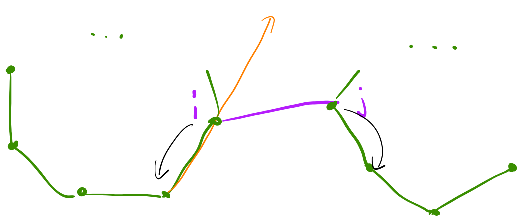
Merge the convex hulls of A and B.

this is the tricky part.

IDEA:



Pseudocode:



While right turn at i or right turn at j:

If right turn at i:

decrement i

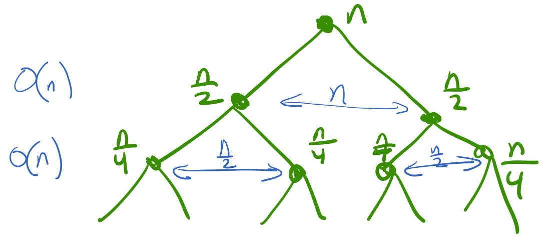
Else:

increment j

COMPLEXITY OF RECURSIVE HULL ALG:

start with n points

we can divide n points in half $\log_2(n)$ times



$\log_2(n)$

each row is merged with $O(n)$ operations

total complexity:

$O(n \cdot \log_2 n)$

$= O(n \log n)$

$\log_2 n = \frac{\log_b n}{\log_b 2}$

OPTIMAL COMPLEXITY FOR 2D HULL ALGORITHMS

Sorting a list of n points: optimal algorithms require $\Omega(n \cdot \log n)$ operations

at least proportional to $n \cdot \log n$

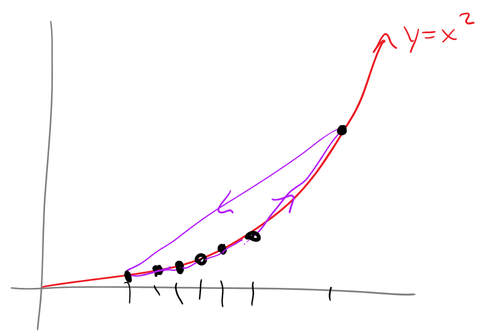
Convex hull algorithms also require $n \cdot \log n$ operations, since if we could find a hull faster, then we could sort faster, which would be a contradiction.

Let $(x_1, x_2, \ldots, x_n)$ be a list of numbers (unsorted).

Construct a list of points $\{(x_1, x_1^2), (x_2, x_2^2), \ldots (x_n, x_n^2)\}$

Find the convex hull of these points.

This returns a list of the hull vertices, which correspond to my original points, in order.



3D Convex Hulls

We need to consider points, edges, and <u>faces</u>.

in memory: list of points, specified by xyz-coordinates
list of triangular faces, each specified by 3 indexes of points

If I have a set S of n points in 3D:
then $\text{conv}(S)$ includes less than $3n$ edges
and less than $2n$ faces.

combinatorial complexity of a 3D convex hull is $O(n)$

ALGORITHMS FOR 3D CONVEX HULLS

- Incremental alg: generalizes to 3D $O(n^2)$
- Gift wrapping alg: generalizes to 3D $O(n \cdot f)$
- Graham scan alg: ???
- Divide and conquer: generalize to 3D $O(n \cdot \log n)$