Lecture 2: Divide and Conquer

- Paradigm
- Convex Hull
- Median finding

Paradigm

Given a problem of size n divide it into subproblems of size $\frac{n}{b}$, $a \geq 1$, $b > 1$. Solve each subproblem recursively. Combine solutions of subproblems to get overall solution.

$$T(n) = aT(\frac{n}{b}) + [\text{work for merge}]$$

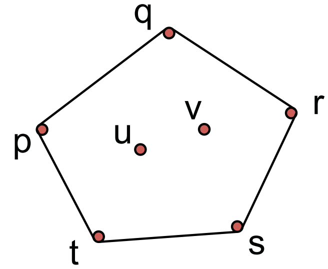
Convex Hull

Given n points in plane

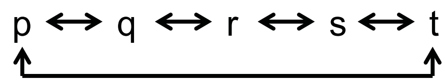
$$S = \{(x_i, y_i) | i = 1, 2, \ldots, n\}$$

assume no two have same x coordinate, no two have same y coordinate, and no three in a line for convenience.

Convex Hull (CH(S)): smallest polygon containing all points in S.



CH(S) represented by the sequence of points on the boundary in order clockwise as doubly linked list.



Brute force for Convex Hull

Test each line segment to see if it makes up an edge of the convex hull

- If the rest of the points are on one side of the segment, the segment is on the convex hull.
- else the segment is not.

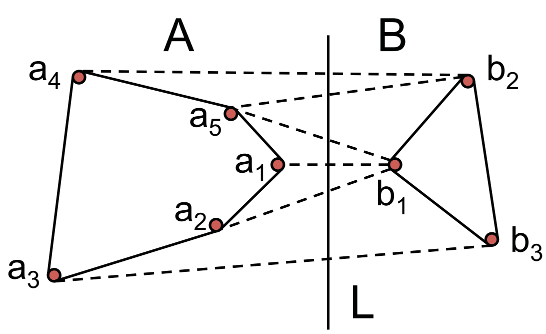
$O(n^2)$ edges, $O(n)$ tests $\Rightarrow O(n^3)$ complexity
Can we do better?

Divide and Conquer Convex Hull

Sort points by x coord (once and for all, $O(n \log n)$)
For input set S of points:

- Divide into left half A and right half B by x coords
- Compute $CH(A)$ and $CH(B)$
- Combine CH's of two halves (merge step)

How to Merge?



- Find upper tangent (a_i, b_j). In example, (a_4, b_2) is U.T.
- Find lower tangent (a_k, b_m). In example, (a_3, b_3) is L.T.

- Cut and paste in time $\Theta(n)$.

First link a_i to b_j, go down b ilst till you see b_m and link b_m to a_k, continue along the a list until you return to a_i. In the example, this gives (a_4, b_2, b_3, a_3).

Finding Tangents

Assume a_i maximizes x within $CH(A)$ $(a_1, a_2, \ldots, a_p)$. b_1 minimizes x within $CH(B)$ $(b_1, b_2, \ldots, b_q)$

L is the vertical line separating A and B. Define $y(i,j)$ as y-coordinate of intersection between L and segment (a_i, b_j).

Claim: (a_i, b_j) is uppertangent iff it maximizes $y(i,j)$.

If $y(i,j)$ is not maximum, there will be points on both sides of (a_i, b_j) and it cannot be a tangent.

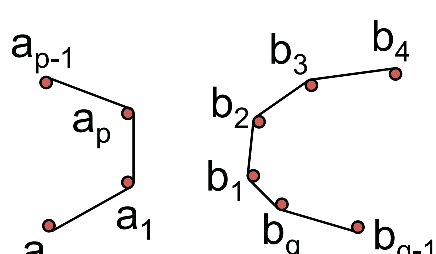
Algorithm: Obvious $O(n^2)$ algorithm looks at all a_i, b_j pairs. $T(n) = 2T(n/2) + \Theta(n^2) = \Theta(n^2)$.

i = 1
j = 1
while $(y(i, j+1) > y(i,j)$ or $y(i-1, j) > y(i,j))$
    if $(y(i, j+1) > y(i,j))$ ▷ move right finger clockwise
        $j = j + 1 (\mod q)$
    else
        $i = i - 1 (\mod p)$ ▷ move left finger anti-clockwise
    return (a_i, b_j) as upper tangent

Similarly for lower tangent.

$$T(n) = 2T(\frac{n}{2}) + \Theta(n) = \Theta(n \log n)$$

Intuition for why Merge works



a_1, b_1 are right most and left most points. We move anti clockwise from a_1, clockwise from b_1. $a_1, a_2, \ldots, a_q$ is a convex hull, as is $b_1, b_2, \ldots, b_q$. If a_i, b_j is such that moving from either a_i or b_j decreases $y(i, j)$ there are no points above the (a_i, b_j) line.

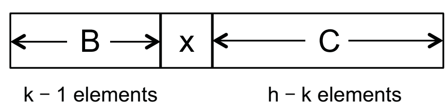
The formal proof is quite involved and won't be covered.

Median Finding

Given set of n numbers, define $rank(x)$ as number of numbers in the set that are $\leq x$. Find element of rank $\lfloor \frac{n+1}{2} \rfloor$ (lower median) and $\lceil \frac{n+1}{2} \rceil$ (upper median).

Clearly, sorting works in time $\Theta(n \log n)$.

Can we do better?



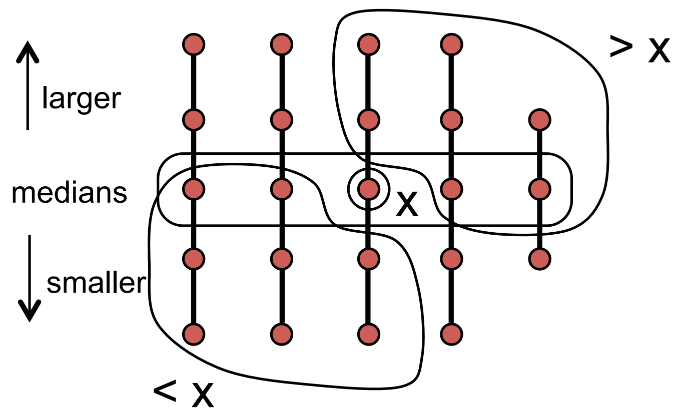
SELECT(S, i)

Pick $x \in S$ ▷ cleverly
Compute $k = rank(x)$
$B = \{y \in S | y < x\}$
$C = \{y \in S | y > x\}$
if $k = i$
    return x
else if $k > i$
    return Select(B, i)
else if $k < i$
    return Select$(C, i - k)$

Picking x Cleverly

Need to pick x so $rank(x)$ is not extreme.

- Arrange S into columns of size 5 ($\lceil \frac{n}{5} \rceil$ cols)
- Sort each column (bigger elements on top) (linear time)
- Find "median of medians" as x



How many elements are guaranteed to be $> x$?

Half of the $\lceil \frac{n}{5} \rceil$ groups contribute at least 3 elements $> x$ except for 1 group with less than 5 elements and 1 group that contains x.

At lease $3(\lceil \frac{n}{10} \rceil - 2)$ elements are $> x$, and at least $3(\lceil \frac{n}{10} \rceil - 2)$ elements are $< x$

Recurrence:

$$T(n) = \begin{cases} O(1), & \text{for } n \le 140 \\ T(\lceil \frac{n}{5} \rceil) + T(\frac{7n}{10} + 6), \Theta(n), & \text{for } n > 140 \end{cases} \tag{1}$$

Solving the Recurrence

Master theorem does not apply. Intuition $\frac{n}{5} + \frac{7n}{10} < n$.

Prove $T(n) \le cn$ by induction, for some large enough c.

True for $n \le 140$ by choosing large c

$$T(n) \le c\lceil \frac{n}{5} \rceil + c(\frac{7n}{10} + 6) + an \tag{2}$$

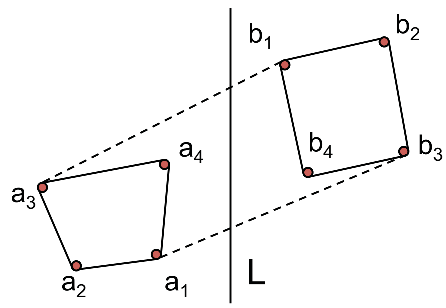
$$\le \frac{cn}{5} + c + \frac{7nc}{10} + 6c + an \tag{3}$$

$$= cn + (-\frac{cn}{10} + 7c + an) \tag{4}$$

If $c \ge \frac{70c}{n} + 10a$, we are done. This is true for $n \ge 140$ and $c \ge 20a$.

Appendix 1

Example



a_3, b_1 is upper tangent. $a_4 > a_3$, $b_2 > b_1$ in terms of Y coordinates.
a_1, b_3 is lower tangent, $a_2 < a_1$, $b_4 < b_3$ in terms of Y coordinates.

a_i, b_j is an upper tangent. Does not mean that a_i or b_j is the highest point. Similarly, for lower tangent.