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.

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

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.

\[ O(n^2) \text{ edges, } O(n) \text{ tests } \Rightarrow O(n^3) \text{ complexity} \]

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. $T(n) = 2T(n/2) + \Theta(n^2) = \Theta(n^2)$.

1. $i = 1$
2. $j = 1$
3. while $(y(i, j + 1) > y(i, j)$ or $y(i - 1, j) > y(i, j))$
4.  
5.  
6. else
7.  
8. 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 $\lceil \frac{n+1}{2} \rceil$ (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)
1    Pick $x \in S \triangleright$ cleverly
2    Compute $k = rank(x)$
3    $B = \{y \in S | y < x\}$
4    $C = \{y \in S | y > x\}$
5    if $k = i$
6        return $x$
7    else if $k > i$
8        return Select($B, i$)
9    else if $k < i$
10       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 ($\lceil \frac{5}{5} \rceil$ 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 $\left\lceil \frac{n}{5} \right\rceil$ groups contribute at least 3 elements $> x$ except for 1 group with less than 5 elements and 1 group that contains $x$.

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

Recurrence:

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

Solving the Recurrence

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

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

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

$$T(n) \leq c\left\lceil \frac{n}{5} \right\rceil + c\left( \frac{7n}{10} + 6 \right) + an \leq \frac{cn}{5} + c + \frac{7nc}{10} + 6c + an = cn + \left( -\frac{cn}{10} + 7c + an \right)$$

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

5
Appendix 1

Example

\[ a_3, b_1 \text{ is upper tangent. } a_4 > a_3, b_2 > b_1 \text{ in terms of Y coordinates.} \]
\[ a_1, b_3 \text{ is lower tangent, } a_2 < a_1, b_4 < b_3 \text{ in terms of Y coordinates.} \]

\[ a_i, b_j \text{ is an upper tangent. Does not mean that } a_i \text{ or } b_j \text{ is the highest point.} \]
Similarly, for lower tangent.