1/19/98 - day 3

1 Merge Convex hull

1. Sort $P_1, ..., P_n$ by the X value

MCH($P_1, ..., P_n$)

1. H_1= MCE($P_1, ..., P_{n/2}$), H_2= MCE($P_{n/2+1}, ..., P_n$)
2. Merge Hull(H_1, H_2)

1.1 Upper Hull and Lower Hull

The convex hull algorithm can be built to find the upper hull and the lower hull then the upper and lower hulls can be easily integrated.

upper hull: The portion of the points defining the convex hull which spans from the point with the smallest x value to the point with the largest x value containing the point with the largest y value.

1.2 Upper Merge Hull

Merge upper hull(H_1, H_2) = start with the inner (x value near the median point) points, throwing out those with a y value lower than the maximum.

1.3 Timing analysis

$T(n) = 2T(n/2) + c(h_1 + h_2)$ which implies $T(n) = O(nlg(n))$

2 Quick Hull

QUH($l, P = \{P_1, ..., P_n\}, r$) =

1. find highest point (large y value) m from line (l, r)
2. remove all points below line (l, r)
3. QUH(l, P, m),QUH(m, P, r)

2.1 Worst case

The worse case is when every point is on an upper semicircle, causing the time equation

$T(n) = T(n-1) + cn$ which means $T(n) = \Omega(cn^2)$

2.2 average case

The average case is points distributed uniformly throughout a semicircle. It will have time complexity of about: $T(n) = T(n_1) + T(n_2) + cn$ for $n_1 + n_2 \leq \alpha n$ with $\alpha < 1$.

3 The ultimate 2d convex hull

Assume we have a $O(n)$ time median algorithm.
Upper Hull(l, P, r)

1. using median partition P into P_L, P_R based on the x value
2. Find Bridge b $= (a, b)$ in upper hull from P_L, P_R
3. Remove points below the line segments defined by (l, a, b, r) from P_L, P_R
4. return Upper Hull(l, P_L, a), Upper Hull(b, P_R, r)

3.1 Timing

let $h = \#$ edges on hull
The time complexity is $c * n * lg_2(h)$.
Check this by plugging it into the time complexity equation.
$T(n, h) = T(n/2, h_1) + T(n/2, h_2) + cn \leq c * n/2 * (lg_2 h_1 + lg_2 h_2) + c * n$
$\leq c * n/2 * lg_2(h^2/4) + cn = c * n * lg_2(n)$

3.2 finding the bridge

Input: points P partitioned into P_L, P_R

Output: Bridge from P_L to P_R

Support line: line brought in from infinity which just touches the outermost point of a set of points.

1. form $n/2$ pars of points from P and compute their slopes.
2. Find median slope, S.
3. Find support line, SL, with slope S.
4. If SL too steep, throw out left point from each pair s.t. the slope greater than S, recurse.
5. Else, if SL too shallow, throw out right point from each pair s.t. the slope $<$ S, recurse.
6. Else if SL has the "right" slope you are done

3.2.1 Timing analysis

$n/4$ points will be thrown out in each recursion.
$T(n) = T(3/4 * n) + cn$ which has solution: $T(n) = 4cn$