Last time

- Convex Hull
- Brute Force
- Inside(q,abc)
- Intersect(ab,cd)
- Orient(a,b,c).

Today:

- More efficient convex hull algorithms and absolute bounds.
- Two (of many) algorithms.
- One (of one) interesting reduction.
Hanging Chads

- At the end of the last class, we talked about a brute force method of finding all extreme points. Points are extreme with respect to some direction. What directions do we have to test?
• How many different orders can the points a, b, c, d appear in?
• Upper bound is 4!
• Actually is:
  - Abcd
  - Bacd
  - Bcad
  - Bcda
  - Bdca
  - Dbca
  - Dcba
  - (which is reverse of the original order).
• Order only changes when line direction is perpendicular to segment between points.
• So extreme values change only when direction is perpendicular to segments.
• So we can find all mins and maxes by sorting along the $O(n^2)$ directions perpendicular to the segments.
Brute-Force Approach
(instead of extreme points, extreme edges)

for each pair of points \( p, q \in P \) do
if all other points lie on one side of line passing thru \( p \) and \( q \) then keep edge \((p, q)\)

\( O(n^2) \) pairs of points
\( O(n) \) work for each

Total of \( O(n^3) \) to find all convex hull edges.

Then “sort” them to get into CCW order w/ \( O(n \log n) \) additional work.
HW problem 1.3

• Given an unordered list of \( k \) ordered pairs, that have form \[ \{(a_1, a_2) (a_2,a_3) (a_3, a_4)...(a_k,a_1)\}, \]

**ALG:**
Sort list by first element of ordered pair.
let \((x_0,y_0)\) be the first ordered pair.
set \(x = x_0, \) and \(y=y_0, \) and output\((x,y)\)
while \(y != x_0\) do
   Search list for ordered pair \((y,z)\)
   \(x = y\)
   \(y = z\)
   output\((x,y)\)
do

\[ \text{O}(n \log n) \]
\[ \text{O}(n) \text{ loop executions} \]
\[ \text{O}(\log n) \text{ binary search} \]
“HW” problem 1.1a

• Prove the intersection $C$ of 2 convex objects $A,B$ is convex.

• Let $x,y$ be any 2 points in $C$
  - $x$ and $y$ are in both $A$ and $B$ (by definition of intersection).
  - All points between $x$ and $y$ are in $A$ (because $A$ is convex), and in $B$ (because $B$ is convex).
  - Therefore all points between $x,y$ are in the intersection of $A,B$

• Since the linear combination of any 2 points in $C$ is also in $C$, $C$ is convex.

• Yata!, Q.E.D., or $\Box$
HW Problem 1.1b

“The smallest perimeter polygon P containing a set of points S is convex.”

Proof: (By contradiction) Suppose P is the smallest perimeter polygon containing set of points S and it is not convex.

Idea: Use the fact that P is not convex, and construct a new polygon with shorter perimeter.

Let x, y be a pair of points such that the line between them goes outside of P. x, y must exist if P is not convex.
Let x’, y’ be the intersections of the line x,y and the polygon P.

Let P’ be the polygon which uses the edge x’, y’
P’ contains the set of points S because it contains P and P contains S.
P’ has a smaller perimeter than P because it goes directly from x’ to y’.

This contradicts that P is the smallest perimeter polygon. Therefore, the smallest perimeter polygon containing a set of points in convex.

QED, Yata, or “box”.
Show any convex set \( C \) containing \( S \) contains the smallest perimeter polygon \( P \)

**Proof (by contradiction):** Assume that \( C \) is a convex set containing \( S \), and that \( P \) is the smallest perimeter polygon containing \( S \), but that \( C \) does not contain \( P \).

Idea: If \( P \) that isn't in the convex set, we can make a smaller \( P \).

- **First:** there is a vertex \( v \) of \( P \) that is not in \( C \).
  - If every vertex \( v \) of \( P \) was in \( C \), then since \( C \) is convex every edge connecting two vertices would also be in \( C \).
- \( v \) is not part of \( S \), since \( C \) contains \( S \).
- Define \( v' \) by moving epsilon clockwise, and \( v'' \) by moving epsilon ccw.
- if epsilon is small enough, then the triangle \( v,v',v'' \) contains no points in \( S \).
- the length of the edge \( v',v'' \) is less than the sum of the lengths of the edges \( v'v \) and \( vv'' \). So we could shorten the perimeter of \( P \) by replacing \( v \) with \( v'' \) and \( v' \) and we would still have a polygon containing \( S \).

This contradicts that \( P \) is the smallest perimeter polygon containing \( S \).

Therefore the smallest perimeter polygon \( P \) is in any convex set \( C \) containing \( S \).

"box", QED, or Yata!
Jarvis March (gift wrapping)

Alg:
• Create (virtual) $p_0$ to be infinitely far to the left.
• Define $p_1$ to be bottom most point.
• Repeat
  - Define $p_{i+1}$ to be the point with the smallest left turn angle to the line $p_{i-1}, p_i$
  - $i = i + 1$
• Until $p_i == p_1$.
• Output $(p_1... p_i)$

Time?
Graham Scan - Algorithm

GrahamScan(point set P)

let $p_0$ be the point with the minimum y-coordinate

let $<p_1, \ldots, p_m>$ be the remaining points in P, sorted by the angle in counterclockwise order around $p_0$

$Top(S) \leftarrow 0$

$Push(p_0, S); Push(p_1, S); Push(p_2, S)$

for $i \leftarrow 3$ to $m$ do

while the angle formed by points $NextToTop(S), Top(S)$ and $p_i$ makes a non-left turn do

$Pop(S)$

end

$Push(p_i, S)$

end

return $S$

Correctness:

The output is convex, because every turn is a left turn.

Every point not on the convex hull (every point left out) is removed by the "Pop" that point $q$ is inside the triangle "NextToTop, q, $p_i$, so it is not extreme."
Claim: Convex hull is related to sorting (normal lists of numbers)

- Quicksort
- Selection Sort
- Merge Sort
- Bubble Sort?
- Radix Sort?
- * special case. All other sorts just use binary comparisons. Let’s stick to that.
Quick Hull – recursive algorithm

Select two extreme points, minX and maxX

These points are on exterior points
Lower Bound

• Is $\Theta(n \lg n)$ the lower bound running time of any 2D convex hull algorithm?

• Yes
  - Proven by Yao (1981) using decision tree

• Can be proven by reducing sorting to finding convex hull
  - Sorting has lower bound $\Theta(n \lg n)$
Lower Bound

- We can use a convex hull algorithm to sort (Shamos, 1978)

  for each input number \( x \) do
      create a 2D point \((x, x^2)\)
  Construct a hull for these points
  Find the lowest point on the hull and follow the vertices of the hull
• If we can compute hull faster than $\Theta(n \lg n)$ then we can sort faster than $\Theta(n \lg n)$. Impossible!
Merge Sort (divide and conquer)

Sort the points from left to right
Let $A$ be the leftmost $\lceil n/2 \rceil$ points
Let $B$ be the rightmost $\lfloor n/2 \rfloor$ points
Compute convex hulls $H(A)$ and $H(B)$
Compute $H(A \cup B)$ by merging $H(A)$ and $H(B)$

• Merging is tricky, but can be done in linear time
Divide & Conquer

- Need to find the upper and lower tangents
- They can be found in linear time
Divide & Conquer

- Find lower tangent
Back to sorting example.

- What lower bound does the sorting reduction actually give?
History of Convex Hull

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Complexity</th>
<th>Author</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brute Force</td>
<td>$O(n^4)$</td>
<td>Anon</td>
<td></td>
</tr>
<tr>
<td>Gift Wrapping</td>
<td>$O(nh)$</td>
<td>Chand &amp; Kapur</td>
<td>1970</td>
</tr>
<tr>
<td>Graham Scan</td>
<td>$O(n \log n)$</td>
<td>Graham</td>
<td>1972</td>
</tr>
<tr>
<td>Jarvis March</td>
<td>$O(nh)$</td>
<td>Jarvis</td>
<td>1973</td>
</tr>
<tr>
<td>QuickHull</td>
<td>$O(nh)$</td>
<td>Eddy, Bykat</td>
<td>1977, 1978</td>
</tr>
<tr>
<td>Divide &amp; Conquer</td>
<td>$O(n \log n)$</td>
<td>Preparata, Hong</td>
<td>1977</td>
</tr>
<tr>
<td>Monotone Chain</td>
<td>$O(n \log n)$</td>
<td>Andrew</td>
<td>1979</td>
</tr>
<tr>
<td>Incremental</td>
<td>$O(n \log n)$</td>
<td>Kallay</td>
<td>1984</td>
</tr>
<tr>
<td>Marriage-before-Conquest</td>
<td>$O(n \log h)$</td>
<td>Kirkpatrick, Seidel</td>
<td>1986</td>
</tr>
</tbody>
</table>

Next class...