Convex Hulls

Given a set of points \((x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\), the convex hull is the smallest convex polygon containing all the points.

1. Find left-most pt \(A\), add to CH
2. let \(L\) = vertical half-line starting at \(A\)
3. do:
4. for every pt \(B\), find the angle of \(L\) w.r.t. \(AB\)
5. let \(C\) be the pt w. the smallest such angle
6. add \(C\) to the CH (if \(C\neq A\))
7. let \(L\) be the half-line starting at \(C\), in the direction of \(AC\) and \(A=C\)
8. while \(\text{start} \neq A\)

Run time: worst case \(O(n^2)\) if \(k\) pts on the CH: \(O(k\cdot n)\)
Convex Hulls

*Gift-wrapping algorithm* running in time $O(n^2)$, or, more precisely, $O(nk)$ where $k$ is the number of vertices on the hull.

An $O(n \log n)$ algorithm?
Convex Hulls

The **Graham Scan** algorithm

1. find left-most pts $\rightarrow A$
2. sort all pts by angle wrt A (A comes first)
3. add pts $1, 2, 3$ to initially empty CH
4. for i = 4 to n:
   5. add pt i to CH
5. if angle of last three pts in CH is $> 180$:
   6. remove the middle pt from CH
8. return CH

Running time?

$O(n \log n)$
Convex Hulls

A divide-and-conquer algorithm?

Split the pts into \( \frac{n}{2} \) and \( \frac{n}{2} \)
how? e.g. sort the pts by their x-coord.
and take the first \( \frac{n}{2} \) and
the last \( \frac{n}{2} \)

The `div-conq-CM` function:

\[
O(3) \{ \\
1. \text{ if } n \leq 3, \text{ return the triangle formed by these pts (disconnected)} \\
2. \text{ let left pts = the first } \left\lfloor \frac{n}{2} \right\rfloor \text{ pts where } n = 1 \text{ pts} \\
3. \text{ let right pts = the rest} \\
4. \text{ let LCH = div-conq-CM(left pts)} \\
5. \text{ let RCH = div-conq-CM(right pts)} \\
6. \text{ return combine(LCH, RCH)}
\]

Running time?

\[
T(n) = \begin{cases} \\
&c \quad \text{if } n \leq 3 \\
2T(\frac{n}{2}) + c \cdot n \\
\end{cases} \Rightarrow T(n) = O(n \log n)
\]
Convex Hulls

A divide-and-conquer algorithm?

Goal: given two convex hulls, disjoint, combine them into 1 CH in \(O(n)\) time

Running time?

1. Find bottom pt on the LCH \(A\), \(RCM = B\)
2. need to check if \(AB\) connection is OK while
3. if \(AB\) passes through one of the \(CH\) (suppose right), then
   - if previous pt to \(B\) is on the outside of \(AB\), move from \(B\) to previous

Idea (sketch):

1. Combine bottom \(O(n)\)
2. \(i\) if \(AB\) forms \(A > 180^\circ\) angle w. blue (original \(CH\)), suppose on the left, then move \(A\) to previous pt on the LCH
3. Note: need to sweep steps 3 & 4 and repeat until \(AB\) connection OK