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.

Output

\[ A_6, A_7, A_8, A_9, A_{10}, A_{11}, A_{12}, \ldots \]

gift-wrapping algo:

1. Start w. the lowest point \(A\)
2. Find the point \(w\) leftmost dere from \(A\)
3. Let the new point be \(B\)
4. Repeat steps 1-4 with \(A = B\) until \(B\) is the
topmost point

Run. time: \(O(n^2)\) in the worst case
\(O(kn)\) where \(k = \#\) point on the hull
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) sort the points by their angle with respect to a selected point inside the convex hull (e.g., the average of the points)

2) starting at the right-most/topmost point (definitely on the hull), go linearly through the points in the sorted order:
   - connect the last point on the hull with the new point → if angle with respect to the last edge on the hull is convex, keep the point in the hull
   - if not convex:
     - remove the last point on the hull, keep doing this while the angle is not convex

**Note:** in the example, points 2,3 were removed because of 4

---

Running time?

\[ O(n \log n) \rightarrow O(n \log n) \text{ for sorting} \]
\[ O(n) \text{ for scan} \]

\[ \text{see previous slide} \]
Convex Hulls

A divide-and-conquer algorithm?

Idea:
- Initially, sort the points by their x-axis
  (done only once, not in the recursive procedure)

Recursively:
1) split points into the \( \frac{n}{2} \) on the left
   and the \( \frac{n}{2} \) on the right
2) recursively find convex hulls for
   the left points and the right points
3) merge the two hulls
   - how? (think about how to
     do this in linear time)

Running time?

Recurrence: \( T(n) \leq 2T\left(\frac{n}{2}\right) + cn \)

Hence:
- \( O(n \log n) \) for sorting
- \( O(n \log n) \) for \( T(n) \)

\( \{ \) over all \( O(n \log n) \)