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.

**Idea 1:** For every pair of points:
- Check if points on both sides of their connection (requires loop through all other points)
  - if not, then include the connection in the CH

**RUNNING TIME:** \(O(n^3)\)

**Idea 2:**
- **Min. point of hull**: Choose next angle (with the last line)
- **Choose next angle**: etc.
**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 points by their angle to a center point (polar sorting)
2. find the min x coord., let the corresp. point be #1
3. include points #1,#2 on the current hull
4. for cpoint = #3 to #n:
   5. while cpoint on the right of the line connecting the last and the second last point on the hull:
   6. remove the last point from the hull
   7. add cpoint to the hull (end of the list)
8. do (*) except using point #1 in place of cpoint

Running time? \( O(n \log n) \) 
- sorting \( O(n \log n) \)
- everything else: \( O(n) \) (bes. adding: \( O(n) \) steps, removing: \( O(n) \) steps)
Convex Hulls

A divide-and-conquer algorithm?
- Assume points sorted by their x-coordinates
- \text{dc-ch}(a_1, \ldots, a_n):
  - if \( n = 3 \):
    - return the points (e.g., counter-clockwise)
  - \( h_1 = \text{dc-ch}(a_1, \ldots, a_{n/2}) \)
  - \( h_2 = \text{dc-ch}(a_{n/2+1}, \ldots, a_n) \)
  - return puttogether(\( h_1, h_2 \))

Running time?
- \( T(n) \leq 2T(n/2) + cn \)
- \( T(n) = O(n \log n) \)