Convex hulls in 2D: Brute force and gift wrapping

The problem: Given a set P of n points in the plane, find their convex hull.

Properties of the convex hull

- A point is on the CH if and only of it is *extreme* (a point p is extreme if there exists a line l through it such that all other points are on or on one side of l).
- An edge is on the CH if and only of it is *extreme* (a line l is extreme if all points in P are on or on one side of it).
- A point p is **not** on the CH if and only if p is contained in the interior of a triangle formed by three other points of P.
- The points with minimum/maximum x-coordinate are on the CH.
- The points with minimum/maximum y-coordinate are on the CH.
- Walking counter-clockwise (ccw) on the boundary of the CH you make only left turns.
- Consider a point p inside the CH. The points on the boundary of the CH are encountered in sorted radial order wrt p.

Algorithm: Brute force

Idea: Find all extreme edges

```
Algorithm BruteForce (input: points $P$)
    • for all distinct pairs of points $(p_i, p_j)$:
        – if edge $(p_i, p_j)$ is extreme, output it as CH edge
```

Questions:

- How do you check if an edge is extreme, and how fast?
- What is the overall running time of Algorithm BruteForce?

Algorithm: Gift wrapping

Idea: start from a point p guaranteed to be on the CH and find the edge pq of the CH starting at p; repeat from q.

```
Algorithm GiftWrapping (input: points $P$)
    • initialize $CH = \{\}$
    • Let $p_0$ be the point with smallest x-coordinate (if more than one, pick right-most).
      CH.append($p_0$).
    • Find the point $p$ with smallest slope wrt $p_0$. CH.append($p$).
    • repeat
          for each point $p'$ ($p'! = p$): compute ccw angle of $p'$ wrt the previous edge on the CH
          let $q$ be the point with smallest such angle
          //claim: edge $(p, q)$ is on the CH, where $p$ is the last point on the CH
          CH.append($q$)
    • until $q == p_0$
```

Questions:

1. Run Gift Wrapping on a set of points and check how it works. Assume no degenerate cases (no collinear points).
2. What is the running time of Algorithm Gift Wrapping? Express the running time as function of n (input size) and k, where k is the output size (in this case, the size of the CH).

   Note: An algorithm whose running time depends on the output size is called an *output-sensitive* algorithm.
3. How big/small can k be for a set of n points? Show examples.
4. What are the best and worst-case bounds for Gift Wrapping?
5. When is GiftWrapping is a good choice?