Algorithm 2:

*Input.* A set $P$ of points in the plane.

*Output.* A list containing the vertices of $CH(P)$ in clockwise order.

1. Sort the points by $x$-coordinate, resulting in a sequence $p_1, \ldots, p_n$.
2. Put the points $p_1$ and $p_2$ in a list $L_{upper}$, with $p_1$ as the first point.
3. for $i \leftarrow 3$ to $n$
   4. do Append $p_i$ to $L_{upper}$.
   5. while $L_{upper}$ contains more than two points and the last three points in $L_{upper}$ do not make a right turn
   6. do Delete the middle of the last three points from $L_{upper}$.

Then we do the same for the lower convex hull, from the right to the left. We remove the first and the last points of the lower convex hull, and concatenates the two lists into tone.

Analysis of Algorithm 2:

1. Proof of correctness
   a. Are the general observations on which the algorithm is based correct?
   b. Does the sorted order matter if two or more points have the same $x$-coordinate?
   c. What happens if there are three or more collinear points, in particular on the convex hull?

2. Efficiency analysis, proof of running time

Assume $n$ input points. Since the sorting step takes $O(n \log n)$ time, the total time of the upper hull algorithm is:

$$O(n \log n) + \sum_{i=3}^{n} O(k_i)$$

Where $k_i$ represents that $k$ points are removed at the $i_{th}$ step.

Considering that each point can be removed at most once from the upper hull, so

$$\sum_{i=3}^{n} O(k_i) \leq n$$

Hence, the total time complexity of the upper hull algorithm is:

$$O(n \log n) + O(n) = O(n \log n)$$