1. Convex hull in the plane

Convex sets and convex hulls in the plane. A set K in the plane is called *convex* if for every two points $p, q \in K$ the line segment pq is contained in K.

FIGURE 1.1 Convex and nonconvex sets

Every point r of the line segment pq can be written as a *convex combination* of the points p and q:

$$r = \lambda p + (1 - \lambda)q, \qquad \text{where } \lambda \in [0, 1].$$

For coordinates (r_x, r_y) of the point r it means that

$$\begin{aligned} r_x =& \lambda p_x + (1 - \lambda)q_x, \\ r_y =& \lambda p_y + (1 - \lambda)q_y. \end{aligned}$$

The intersection of convex sets is apparently a convex set. The *convex hull* $\mathcal{CH}(P)$ of a set P in the plane is the smallest convex set containing the set P, which can be written in the following way:

$$\mathcal{CH}(P) = \bigcap_{K \supseteq P \text{ convex}} K.$$

We will deal with algorithms which find the convex hull of a finite set P effectively. In this case we use the fact that the convex hull is

$$\mathcal{CH}(P) = \bigcap_{H \supseteq P \text{ a halfplane}} H.$$

We do not need to consider all such halfplanes. It is enough to intersect only those halfplanes that are determined by a pair of points from P and that also contain all the other points from P. Therefore the convex hull of a finite nonempty set with at least two points is a convex polygon which can degenerate to a line segment.

Naive algorithm. In this section we show that only knowledge of mathematical definition does not suffices to creation of a good algorithm. Our naive algorithm will search for oriented segments pq which form the clockwise boundary of the convex hull. It is based on the fact that for such an oriented segment there is no point of the set P lying to the left of the oriented line pq.

FIGURE 1.2 To the left of the oriented line pq there is no point of the set P.

The fact that a point r lies to the left of the oriented line pq can be expressed mathematically by the following condition on the determinant the rows of which are the coordinates of the vectors $q - p$ a $r - p$:

$$\det \begin{pmatrix} q_x - p_x & q_y - p_y \\ r_x - p_x & r_y - p_y \end{pmatrix} > 0.$$

This determinant of the 2×2 matrix can be written also as a determinant of the following 3×3 matrix

$$\det \begin{pmatrix} 1 & p_x & p_y \\ 1 & q_x & q_y \\ 1 & r_x & r_y \end{pmatrix}$$

ALGORITHM 1 from pseudo.pdf

The time complexity of this algorithm for n points in the entry is $O(n^3)$ since we go through all oriented lines determined by n points and for every such line we test if remaining points lie to the left. The next shortage is the fact that the algorithm is unstable. If three points p_1, p_2, p_3 lie on one line, it can happen that oriented segments p_1p_2, p_2p_3 a p_1p_3 are elements of the set E of boundary segments. However then we can have a problem with ordering them to the sequence in clockwise order.

FIGURE 1.3 Three points of the set P being on only one boundary line segment.

Better algorithm - Graham's scan. We get a much better result if we order the points of the set P in a suitable way before we start with searching boundary segments of the convex hull. We use lexicographic layout of points, first according to the coordinate x and then according to the coordinate y. Explicitly,

$$p < q \quad \text{iff } p_x < q_x \text{ or } (p_x = q_x \text{ and } p_y < q_y).$$

In this arrangement let us denote the points from the smallest to the biggest

$$p_1 < p_2 < p_3 < \cdots < p_{n-1} < p_n.$$

The point p_1 which is the "most left" and the point p_n which is the "most right" have to lie on the boundary of the convex hull and split this boundary into an upper and a lower part. To omit long formulations we will talk (not exactly) about an upper and lower convex hull instead of talking about an upper and a lower part of the boundary of the convex hull.

FIGURE 1.4 The upper (blue) and the lower (green) convex hull.

The clockwise arrangement of the points of the upper convex hull corresponds with their lexicographic order. That is why our algorithm will search for the upper convex hull of the sets

$$P_i = \{p_1, p_2, \ldots, p_i\}$$

from $i = 2$ to $i = n$. Then it will search for the lower convex hull of the sets

$$\{p_{n-i}, p_{n-1+1} \cdots, p_{n-1}, p_n\}$$

for $i = 1$ do $n - 1$ in a similar way.

Let $\mathcal{L}$ be the lexicographically ordered list of the vertices of the upper convex hull of the set P_i. It always contains points p_1 and p_i. Now we show how to change the list

$\mathcal{L}$ to contain the vertices of the upper convex hull of the set P_{i+1}. The lexicografically biggest point p_{i+1} has to be in the upper convex hull, so we start by adding it into $\mathcal{L}$.

The points which do not belong to the upper convex hull of the set P_{i+1} will be discarded from the list $\mathcal{L}$ gradually in the following way: We take the last three points $p_j < p_i < p_{i+1}$ in the list $\mathcal{L}$. If the point p_{i+1} lies to the right of the oriented line $p_j p_i$, we will say that the points p_j, p_i a p_{i+1} form a *right turn*. In this case we will not change the list $\mathcal{L}$ since the points of the set P_{i-1} lie to the right of the oriented line $p_i p_{i+1}$. So we have the list of vertices of the upper convex hull of the set P_{i+1} and we can start to search for the list of vertices of the upper convex hull for the set P_{i+2}, if $i + 2 \leq n$.

FIGURE 1.5 Right turn.

If the points p_j, p_i a p_{i+1} do not form a right turn, we exclude the middle one, i. e. p_i, from the list $\mathcal{L}$. If p_i lies on the segment $p_j p_{i+1}$, we do not need it for the description of the upper convex hull (although it lies in it). If p_i lies to the right of the oriented line $p_j p_{i+1}$, it is not in the upper convex hull of the set P_{i+1} and has to be discarded.

FIGURE 1.6 Cases in which the last three points of the list $\mathcal{L}$ do not form a right turn.

After discarding the point p_i from the list $\mathcal{L}$ we take again the last three points of this list, if they exist, and we carry out the same procedure with them. We repeat this process until

- either only two points remain in the list $\mathcal{L}$,
- or the last three points in the list $\mathcal{L}$ form a right turn.

This completes the searching for the list of vertices of the upper convex hull of the set P_{i+1}. We illustrate this process using the following animation.

ANIMATION 1.1 Texts:

- We take $\mathcal{L} = \{p_1, p_2\}$ and add p_3 to $\mathcal{L}$. The points p_1 p_2 and p_3 form a right turn.
- We add p_4 to the list $\mathcal{L} = \{p_1, p_2, p_3\}$. The points p_2, p_3 and p_4 form a right turn.
- We add p_5 to the list $\mathcal{L} = \{p_1, p_2, p_3, p_4\}$. The points p_3, p_4 and p_5 do not form a right turn.
- We discard the point p_4 from the list $\mathcal{L}$. The points p_2, p_3 and p_5 do not form a right turn.
- We discard the point p_3 from the list $\mathcal{L}$. The points p_1, p_2 and p_5 form a right turn.
- We add p_6 to the list $\mathcal{L} = \{p_1, p_2, p_5\}$. The points p_2, p_5 and p_6 form a right turn.
- We add p_7 to the list $\mathcal{L} = \{p_1, p_2, p_5, p_6\}$. The points p_5, p_6 and p_7 do not form a right turn.

- We discard the point p_6 from the list $\mathcal{L}$. The points p_2, p_5 and p_7 form a right turn.
- We add p_8 to the list $\mathcal{L} = \{p_1, p_2, p_5, p_7\}$. The points p_5, p_7 and p_8 do not form a right turn.
- We discard the point p_7 from the list $\mathcal{L}$. The points p_2, p_5 and p_8 form a right turn.
- We add p_9 to the list $\mathcal{L} = \{p_1, p_2, p_5, p_8\}$. The points p_5, p_8 and p_9 form a right turn.
- $\mathcal{L} = \{p_1, p_2, p_5, p_8, p_9\}$. The upper convex hull is complete.

The vertices of the lower convex hull can be found analogously.

Pseudocode and time complexity. The above verbal description can be formalized by the following pseudocode

ALGORITHM 2 from pseudo.pdf

Theorem 1.1. *The above algorithm is correct and its time complexity depending on the number of points of the set P is $O(n \log n)$.*

Proof. The proof that the algorithm gives us the list of vertices of the convex hull of the set P ordered clockwise reduces to the proof that we get the list of vertices of the upper and lower convex hulls ordered lexicographically. The proof for the upper convex hulls proceeds by induction: We show that the algorithm will create the list for the upper convex hull of the set P_{i+1} from the list of vertices of the upper convex hull of the set P_i. We can carry it out by similar considerations as we did during the verbal description of the algorithm. Practically focused readers will no longer be tired of them. Theoretically focused readers can try to handle the proof by themselves.

As for the time complexity, the first step of the algorithm – to order n points lexicographically takes the time $O(n \log n)$ as it is well known. The rest of the algorithm takes only the time $O(n)$. We put each of the points of the set P into the list $\mathcal{L}$ once, and we take it out at most once. So the time required for the list to be created must be proportional to the number of points.

The reader will surely come up with a number of ways to improve this algorithm. However, none of them improves the estimate $O(n \log n)$. □

Another algorithm. There are a lot of other algorithms for the convex hull of a finite set in the plane. Let us indicate one of them, which is suitable to use if we know in advance that the number of vertices of the convex hull is limited by a number k which is small with respect to n, the number of points of the set P. The process of the algorithm is reminiscent of packaging, so it is called Gift wrapping.

For simplicity, let's assume that no three points lie in a line. First, we find the left most point of the set P (more precisely, the smallest in the above lexicographic layout) and mark it p_1. We make lines from the point p_1 to other points of the set P and consider their slopes. Let p_2 be such a point that p_1p_2 has the biggest slope from all these lines. Now, we link the point p_2 to the remaining $n-2$ points of the set P. We select a point p_3 so that the convex angle $\angle p_1p_2p_3$ is the largest. In this way we proceed until we get back to the point p_1. This is how we wrapped the set P into the convex hull. If the convex hull is a polygon with k vertices, the time complexity of this algorithm is $O(kn)$. In each of the k vertices, we spend $O(n)$ time.

ANIMATION 1.2 Texts:

- We choose the most left point p_1 and make lines from p_1 to the other points. We choose the point p_2.
- We make lines from p_2 to the other points and choose the point p_3.
- We make lines from p_3 to the other points and choose the point p_4.
- We make lines from p_4 to the other points and choose the point p_5. It is equal to p_1.
- We are finished. The vertices of the convex hull in clockwise order are p_1, p_2, p_3 and p_4.