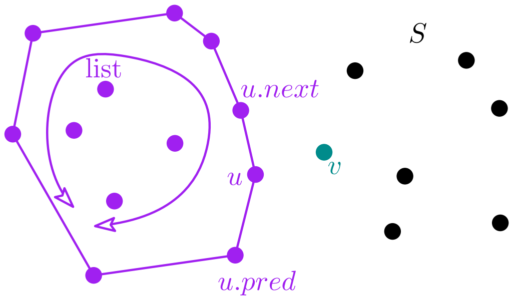
1 Convex hull: definitions, classical algorithms.

1.1 Convex hull by triangulation

Consider the following convex hull algorithm:

```
Input :  S a point set.
sort S by x-coordinate;
create a circular list with the three leftmost points of S
    such that (u, u.next, u.next.next) is positively oriented and u rightmost;
S = S \ {u, u.next, u.next.next};
while S ≠ ∅  {
    v =leftmost point in S;
    S = S \ {v};
    w = copy(u); •
    while (v, u, u.next) negative
        {u = u.next; }•
    v.next = u;  u.pred = v;
    while (v, w, w.pred) positive
        {w = w.pred; }•
    v.pred = w;  w.next = v;
    u = v;
}
```



1.1.1 Run the algorithm

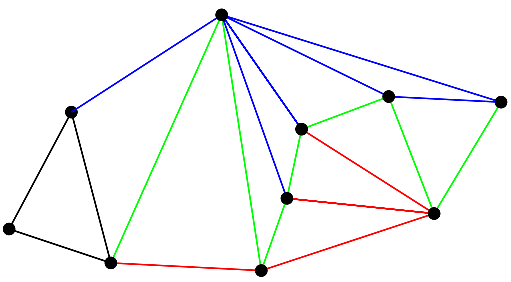
Run (by hand) the algorithm on the example provided on the attached example and draw on the example the line segment uv in green each time the code pass through the line of the code marked with a green dot, draw uv in blue for the code line with a blue dot, and draw wv in red for the code line with a red dot,

1.1.2 Complexity

What is the complexity of this algorithm on a set of n points (hint: use Euler relation).

1.1 Correction:

1.1.1 Run the algorithm



1.1.2 Complexity

Sorting step is $O(n \log n)$.
The complexity of the rest is bounded (up to a constant) by the number of times that the algorithm goes through the lines marked by the three dots. Each time the algorithm goes there, one can trace an edge of a triangulation of the points, thus the complexity is the number of edges in a triangulation.
Compared to the situation during the lecture (3D convex hull), the infinite face is of some unknown size $k \leq n$. Let e be the number of edges, and t the number of triangles. On the one hand, we have $(t+1) - e + n = 2$ (Euler relation). On the other hand, each triangle has three edges, each edge is in two faces: $2e = k + 3t$. One can deduce $e = 3n$.
Thus after the sorting preprocessing, the algorithm is linear.