Выпуклая оболочка.

Гащук Елизавета, 332 группа

1.Постановка задачи. Дан набор точек на плоскости. Требуется вычислить выпуклую оболочку данного множества точек.

2.Определение и псевдокод.

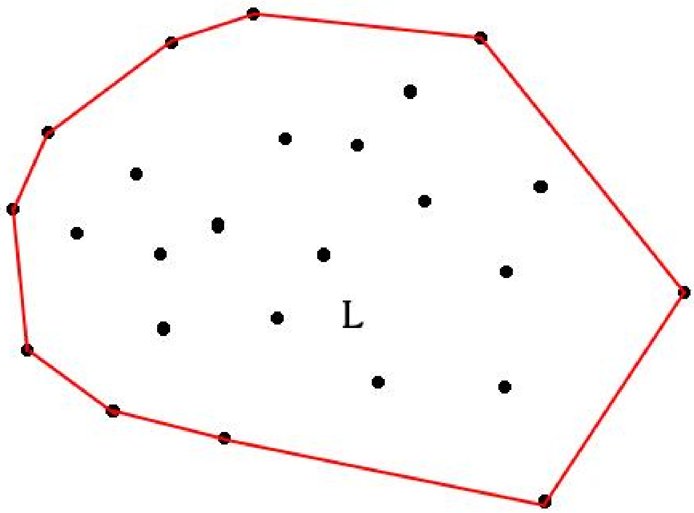
Опр 1. Пусть в d - мерном пространстве E^d заданы n различных точек $\{p_i\}_{i=1}^n$. Множество точек

$p = \alpha_1 p_1 + \alpha_2 p_2 + \ldots + \alpha_n p_n$

$(\alpha_i \in \mathbb{R},\ \alpha_i > 0,\ \alpha_1 + \alpha_2 + \ldots + \alpha_n = 1)$

называется *выпуклым множеством*, порожденным точками $\{p_i\}_{i=1}^n$.

Опр 2. Пусть в d - мерном пространстве E^d заданы n различных точек $\{p_i\}_{i=1}^n = L$. *Выпуклой оболочкой* $Conv(L)$ множества L называется наименьшее выпуклое множество, содержащее L.



Пример $Conv(L)$ множества черных точек L на плоскости

Algorithm 1 Convex Hull

Отсортировать точки по возрастанию координаты x
Отсортировать подпоследовательности, образованные одинаковой координатой x, по возрастанию координаты y. Отсортированный массив - $\{p_i^*\}_{i=0}^{n-1}$
Положить p_0^*, p_1^* в list L_{upper}
for i = 2 **to** n - 1 i++ **do**
    Положить p_i^* в конец list L_{upper}
    while L_{upper} содержит более двух точек **and** последние три точки из L_{upper} не образуют правый поворот **do**
        удалить среднюю точку среди последних трех из L_{upper}
    end while
end for
Положить точки p_{n-1}^* and p_{n-2}^* в лист L_{lower}
for i = n - 3 **to** 0 i- - **do**
    Положить p_i^* в конец list L_{lower}
    while L_{lower} содержит более двух точек **and** последние три точки из L_{lower} не образуют правый поворот **do**
        удалить среднюю точку среди последних трех из L_{lower}
    end while
end for
удалить первую и последнюю точки из L_{lower}, так как они дублируются в L_{upper}
return $L_{upper} + L_{lower}$

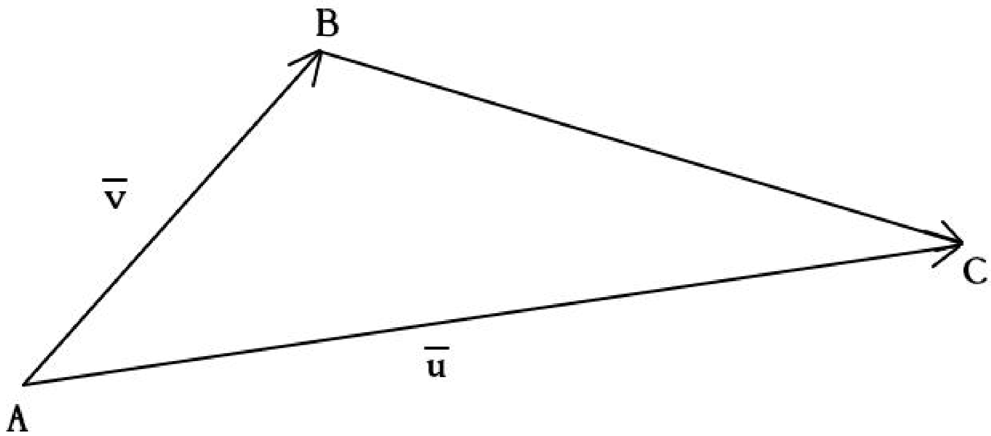
В алгоритме *Convex Hull* необходимо проверять, какой поворот образует тройка точек A, B, C: правый или левый.

Утверждение 1. Дана тройка точек A, B, C $\in \mathbb{R}^2$. $\vec{V} = \vec{AB}$, $\vec{U} = \vec{AC}$. Тогда, если:

- $Sign([\vec{U}, \vec{V}]) = 1$, то тройка точек A, B, C образует правый поворот.
- $Sign([\vec{U}, \vec{V}]) = -1$, то тройка точек A, B, C образует левый поворот.
- $[\vec{U}, \vec{V}] = 0$, то тройка точек A, B, C лежит на одной прямой.

Доказательство:

Очевидно, что при $[\vec{U}, \vec{V}]_3 = 0$ тройка точек A, B, C лежит на одной прямой, так как $|[\vec{U}, \vec{V}]_3|$ – площадь параллелограмма, натянутого на вектора $\vec{V}$ и $\vec{U}$.



Из курса аналитической геометрии знаем, что, если кратчайший поворот от $\vec{U}$ до $\vec{V}$ против часовой стрелки, то $Sign([\vec{U},\vec{V}]) = 1$, если по часовой, получим -1.

Будем постепенно соединять вершины, образуя треугольник. Сначала грань через вершины A и C. Если $Sign([\vec{U},\vec{V}]) = 1$, то чтобы соединить C и B, необходимо повернуть налево относительно направления $\vec{AC}$. Чтобы замкнуть кривую, необходимо соединить B и A, совершив левый поворот относительно направления $\vec{CB}$. Тогда при обратном обходе A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ A все повороты будут правыми.

При $Sign([\vec{U},\vec{V}]) = -1$ аналогично получаем, что все повороты при A $\rightarrow$ B $\rightarrow$ C $\rightarrow$ A – левые.

$\blacksquare$

Сложность алгоритма Convex Hull.

• Сортировка Хоара n точек имеет среднюю сложность $n\log(n)$. Один раз сортируем по x координате $\rightarrow n\log(n)$.

• После сортировки по x - координате получили k подпоследовательностей, образованных точками с одинаковой x - координатой. Скажем, что длина i - ой подпоследовательности есть len_i. Тогда нам нужно еще $\sum_{i=1}^{k} len_i \log(len_i)$ действий.

• Далее n - 2 раза проверяем поворот. Тут требуется $11(n-2)$ действий (для L_{upper}).

• Далее n - 2 раза проверяем поворот. Тут требуется $11(n-2)$ действий (для L_{lower}).

Итого имеем: $n\log(n) + \sum_{i=1}^{k} len_i \log(len_i) + 2{\cdot}11(n-2) = \mathcal{O}(n\log n)$ в среднем.

3.Результаты.