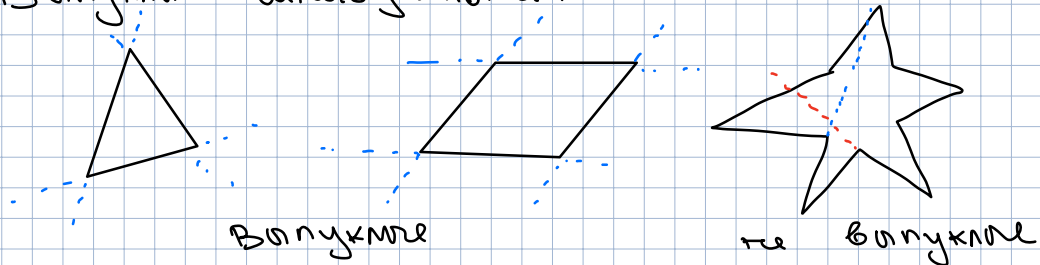


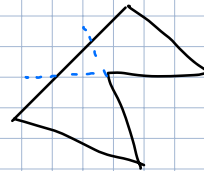
Вогнутые оболочки лекция 27

①

Опр Вогнутый многоугольник

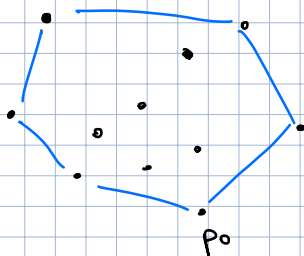


если никакое продолжение стороны не пересекает фигуру.



Опр Вогнутая оболочка.

Есть набор точек 2D пространстве



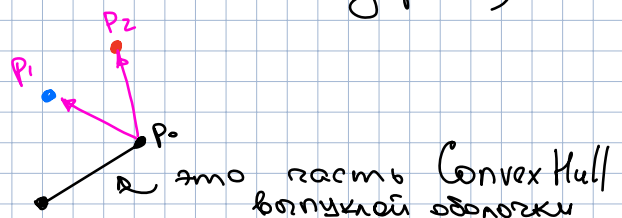
построить
вогнутый
многоугольник
с вершинами
из набора точек

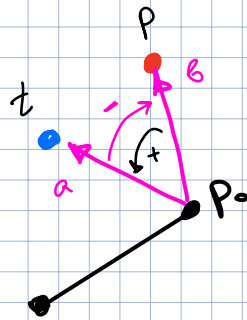
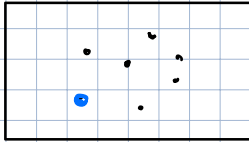
Так, чтобы все
точки набора
лежали внутри
мн-ка.

①

Алгоритм Дира ("Заворачивание подарка")

а) Введение





$$\vec{a} = \vec{P_0 P}$$

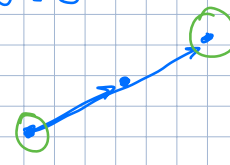
$$\vec{b} = \vec{P_0 t}$$

$a \wedge b > 0 \Rightarrow t$ справа от P

$a \wedge b < 0 \Rightarrow t$ слева от P

$$a \wedge b = 0$$

Все точки
одного полуплоскости.
(никакие три
точки не
лежат на одной
прямой)



Все много
(необходимо
рассмотреть
всего то)

берем вектор,
больший по
модулю.

2) Алгоритм

функция Jarvis (points) {
// массив точек

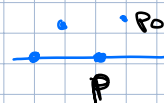
$P_0 = \text{points}[0]$ // указать функцию
 $\text{hull} = []$ (небольшая точка)

for $i = 0 \dots \text{points.size}()$:

$P = \text{points}[i]$

if $P_0.y > P.y$ or
($P_0.y = P.y$ and $P_0.x > P.x$)

$\hookrightarrow P_0 = P$



}

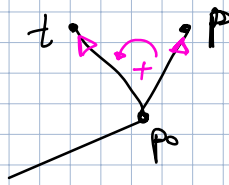
while (true) {

$t = P_0$

for $i = 0 \dots \text{points.size}()$:

$P = \text{points}[i]$

if $(P - P_0) \wedge (t - P_0) > 0$
 $\hookrightarrow t = P$



повтор
h

O(N)

```

    if (t == p0)
        break
    else:
        p0 = t
        hull.push(t)
    }
return hull
}

```

// закончил поиск
внутри оболочки



б) Асимптотика: $T(N) = O(N \cdot h)$
 h - размер Convex Hull

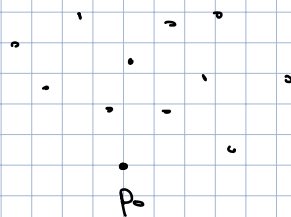
в) Примечание: поиск наивной оболочки
 тогда:
 просто: $N^2 \sim 10^{16}$
 сложенно: $10^{10} + 10^4 \sim 10^{10}$

$N = 10^8$
 $h = 100$



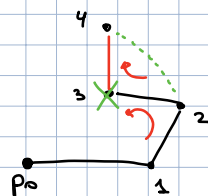
III Алгоритм Грэхема

а) Интерактивный Ракет

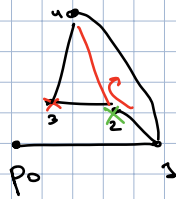


Сортируем ($O(N \log N)$)
 в порядке возрастания
 поперечной длины.

Вогнутая оболочка - последовательность
 последовательности отсортир-х точек.



направление поворота -
 влево.



АТД смех
Convex Hull

8) Асимптотика

- подготовка точек $O(N \log N)$
 $O(N)$ - найти p_0 (все точки лежат по часу)
 $O(N \log N)$ - сортировка

• алгоритм $O(2N) = O(N)$

$$T(N) = O(N \log N) + O(N) = O(N \log N)$$

9) Алгоритм.

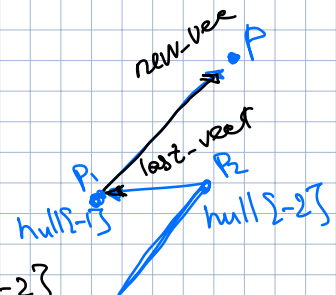
Ф.м. Graham (points)

```
// найти самую левую.
p0 = points[0]
for i = 0... points.size():
    p = points[i]
    if p0.y > p.y or (p0.y == p.y and p0.x > p.x)
        p0 = p
}
sort (points, compare)
```

compare (point a, point b)
return (a-p0)^(b-p0) > 0

hull = [] // std::vector → АТД ГРК.

```
for i = 0... points.size():
    p = points[i]
    while (hull.size() >= 2)
    {
        new-vec = p - hull[hull.size()-1]
        last-vec = hull[hull.size()-1] - hull[hull.size()-2]
```



```

// new_vec > last_vec
// new_vec > last_vec
// new_vec > last_vec
if (new_vec ^ last_vec > 0)
    hull.pop_back();
else:
    break
}
hull.push_back(p)
}
return hull
}

```

$hull[i] = hull[hull.size() - 1]$