计算几何浅谈

rotcar

2023 年 2 月 2 日

目录

FAQ

FAQ

Q：这节课讲什么?

FAQ

Q：这节课讲什么？
A：计算几何。

FAQ

Q：这节课讲什么？
A：计算几何。
Q：在 OI 里常用吗？

FAQ

Q：这节课讲什么?
A：计算几何。
Q：在 OI 里常用吗?
A：不常用。但是毒瘤 N_z_ 模拟赛搬过。

FAQ

Q：这节课讲什么？
A：计算几何。
Q：在 OI 里常用吗？
A：不常用。但是毒瘤 N_z_ 模拟赛搬过。
Q：可以不听吗？

FAQ

Q：这节课讲什么?
A：计算几何。
Q：在 OI 里常用吗?
A：不常用。但是毒瘤 N_z_ 模拟赛搬过。
Q：可以不听吗?
A：这节课会从基础开始讲。实际上你会发现课件里的题大部分是各省省选历年真题，所以还是很有必要的。实在不想听可以 v 我 50。

FAQ

Q：这节课讲什么？
A：计算几何。
Q：在 OI 里常用吗？
A：不常用。但是毒瘤 N_z_ 模拟赛搬过。
Q：可以不听吗？
A：这节课会从基础开始讲。实际上你会发现课件里的题大部分是各省省选历年真题，所以还是很有必要的。实在不想听可以 v 我 50。
Q：讲课人实力怎么样？

FAQ

Q：这节课讲什么？
A：计算几何。
Q：在 OI 里常用吗？
A：不常用。但是毒瘤 N_z_ 模拟赛搬过。
Q：可以不听吗？
A：这节课会从基础开始讲。实际上你会发现课件里的题大部分是各省省选历年真题，所以还是很有必要的。实在不想听可以 v 我 50。
Q：讲课人实力怎么样？
A：讲课人很菜，所以有些地方可能会出错，谅解一下。

Will liuhengxi achieve LGM in grade 9?

By dXqwq, history, 10 days ago,

> As far as I know, he is in grade 9 now, which is one year younger than orzdevinwang.
>
> He is gaining rating in recent contests and previous LGMs in grade 9 are djq-cpp and orzdevinwang(comment below if you know more).
>
> Meanwhile I'm stuck between 2600~2700, hope I will become stronger next year o(>_<)o

+49 | dXqwq | 10 days ago | 5

Comments (5)

Write comment?

N_z__ — 10 days ago, # Edit | +34

As a classmate of liuhengxi,I think he will.

→ Reply

moonrise__ — 10 days ago, # ^ | +68

As a classmate of liuhengxi, I think he wants a girlfriend.

→ Reply

User_Carrot — 10 days ago, # ^ | +20

As a classmate and also a fan of liuhengxi, I think he will. And he wants a girlfriend btw.

→ Reply

聊天记录:我要找妹子 群成员:look 搜索

161条消息记录

【管理员】 looking_for_girlfriend 2023/1/17 10:22:10
@跳跳板 我要找妹子

【管理员】 looking_for_girlfriend 2023/1/17 9:33:34
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/15 22:19:06
今晚 13 点，我要找妹子

【管理员】 looking_for_girlfriend 2023/1/12 15:53:17
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/11 16:35:33
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/10 21:03:34
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/10 20:17:37
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/10 18:56:04
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/9 22:08:06
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 19:42:25
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 19:06:51
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 14:41:06
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 14:17:49
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 14:11:11
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 14:07:55
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 13:41:18
我要找妹子

【管理员】 looking_for_girlfriend 2023/1/8 12:25:41
我要找妹子

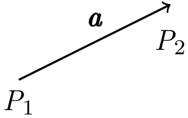
【管理员】 looking_for_girlfriend 2023/1/8 12:25:38
我要找妹子

向量

向量是指有方向的线段。

向量

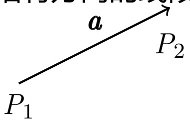
向量是指有方向的线段。

如图：

，这是一个向量。

向量

向量是指有方向的线段。

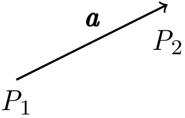
如图：

，这是一个向量。

其中 P_1 和 P_2 是有顺序的，记为 $\boldsymbol{a} = \overrightarrow{P_1P_2}$。

向量

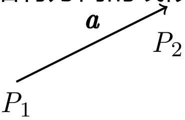
向量是指有方向的线段。



如图：，这是一个向量。
其中 P_1 和 P_2 是有顺序的，记为 $\boldsymbol{a} = \overrightarrow{P_1P_2}$。
该有向线段的长度称为向量的模，记作 $|\boldsymbol{a}|$。

向量

向量是指有方向的线段。

如图：

，这是一个向量。

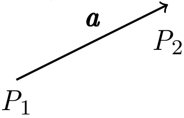
其中 P_1 和 P_2 是有顺序的，记为 $\boldsymbol{a} = \overrightarrow{P_1P_2}$。

该有向线段的长度称为向量的模，记作 $|\boldsymbol{a}|$。

若 P_1 是坐标原点，则向量 $\overrightarrow{P_1P_2}$ 也可记作向量 $\boldsymbol{P_2}$。

向量

向量是指有方向的线段。

如图：

，这是一个向量。

其中 P_1 和 P_2 是有顺序的，记为 $\boldsymbol{a} = \overrightarrow{P_1P_2}$。

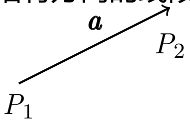
该有向线段的长度称为向量的模，记作 $|\boldsymbol{a}|$。

若 P_1 是坐标原点，则向量 $\overrightarrow{P_1P_2}$ 也可记作向量 $\boldsymbol{P_2}$。

因此向量也有坐标记法，即为当起点是原点时端点的坐标。

向量

向量是指有方向的线段。

如图：

，这是一个向量。

其中 P_1 和 P_2 是有顺序的，记为 $\boldsymbol{a} = \overrightarrow{P_1P_2}$。

该有向线段的长度称为向量的模，记作 $|\boldsymbol{a}|$。

若 P_1 是坐标原点，则向量 $\overrightarrow{P_1P_2}$ 也可记作向量 $\boldsymbol{P_2}$。

因此向量也有坐标记法，即为当起点是原点时端点的坐标。

此外，该向量 $\boldsymbol{a}$ 也可记作 $P_2 - P_1$。

向量的加减法

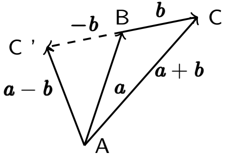
以点 A 为起点，以点 B 为端点作向量 $\boldsymbol{a}$，以点 B 为起点，以点 C 为端点作向量 $\boldsymbol{b}$，则以点 A 为起点，以点 C 为端点的向量称为向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的和，记为 $\boldsymbol{a}+\boldsymbol{b}$。

向量的加减法

以点 A 为起点，以点 B 为端点作向量 $\boldsymbol{a}$，以点 B 为起点，以点 C 为端点作向量 $\boldsymbol{b}$，则以点 A 为起点，以点 C 为端点的向量称为向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的和，记为 $\boldsymbol{a}+\boldsymbol{b}$。
若以点 B 为起点，作 $\overrightarrow{BC'}$，使得 $|\overrightarrow{BC'}| = |\boldsymbol{b}|$，且方向与 $\boldsymbol{b}$ 相反，则以点 A 为起点，以点 C' 为端点的向量称为向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的和，记为 $\boldsymbol{a}-\boldsymbol{b}$。如下图。

向量的加减法

以点 A 为起点，以点 B 为端点作向量 $\boldsymbol{a}$，以点 B 为起点，以点 C 为端点作向量 $\boldsymbol{b}$，则以点 A 为起点，以点 C 为端点的向量称为向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的和，记为 $\boldsymbol{a}+\boldsymbol{b}$。
若以点 B 为起点，作 $\overrightarrow{BC'}$，使得 $|\overrightarrow{BC'}|=|\boldsymbol{b}|$，且方向与 $\boldsymbol{b}$ 相反，则以点 A 为起点，以点 C' 为端点的向量称为向量 $\boldsymbol{a}$ 和向量 $\boldsymbol{b}$ 的和，记为 $\boldsymbol{a}-\boldsymbol{b}$。如下图。



向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$
2. $\boldsymbol{a} \perp \boldsymbol{b} \Leftrightarrow \boldsymbol{a} \cdot \boldsymbol{b} = 0$

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$
2. $\boldsymbol{a} \perp \boldsymbol{b} \Leftrightarrow \boldsymbol{a} \cdot \boldsymbol{b} = 0$
3. $|\boldsymbol{a}|^2 = \boldsymbol{a} \cdot \boldsymbol{a}$

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$
2. $\boldsymbol{a} \perp \boldsymbol{b} \Leftrightarrow \boldsymbol{a} \cdot \boldsymbol{b} = 0$
3. $|\boldsymbol{a}|^2 = \boldsymbol{a} \cdot \boldsymbol{a}$
4. $(\lambda \boldsymbol{a}) \cdot \boldsymbol{b} = \lambda(\boldsymbol{a} \cdot \boldsymbol{b})$

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$
2. $\boldsymbol{a} \perp \boldsymbol{b} \Leftrightarrow \boldsymbol{a} \cdot \boldsymbol{b} = 0$
3. $|\boldsymbol{a}|^2 = \boldsymbol{a} \cdot \boldsymbol{a}$
4. $(\lambda \boldsymbol{a}) \cdot \boldsymbol{b} = \lambda(\boldsymbol{a} \cdot \boldsymbol{b})$
5. $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$

向量的数量积

两个向量的数量积，又称点乘，其结果是一个数。

$$\boldsymbol{a} \cdot \boldsymbol{b} = |\boldsymbol{a}||\boldsymbol{b}| \cos\langle \boldsymbol{a}, \boldsymbol{b} \rangle = x_{\boldsymbol{a}} x_{\boldsymbol{b}} + y_{\boldsymbol{a}} y_{\boldsymbol{b}}$$

其中 $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ 表示两向量 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 的夹角。
其具有如下性质：

1. $\boldsymbol{a} \cdot \boldsymbol{e} = |\boldsymbol{a}| \cos\langle \boldsymbol{a}, \boldsymbol{e} \rangle$
2. $\boldsymbol{a} \perp \boldsymbol{b} \Leftrightarrow \boldsymbol{a} \cdot \boldsymbol{b} = 0$
3. $|\boldsymbol{a}|^2 = \boldsymbol{a} \cdot \boldsymbol{a}$
4. $(\lambda \boldsymbol{a}) \cdot \boldsymbol{b} = \lambda(\boldsymbol{a} \cdot \boldsymbol{b})$
5. $\boldsymbol{a} \cdot \boldsymbol{b} = \boldsymbol{b} \cdot \boldsymbol{a}$
6. $\boldsymbol{a} \cdot (\boldsymbol{b} + \boldsymbol{c}) = \boldsymbol{a} \cdot \boldsymbol{b} + \boldsymbol{a} \cdot \boldsymbol{c}$

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b} \rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b}\rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b}\rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。
其中该式的符号：若 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 的逆时针方向则为正，否则为负。简记为**顺负逆正**。

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b}\rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。
其中该式的符号：若 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 的逆时针方向则为正，否则为负。简记为**顺负逆正**。
可以看出向量积的如下性质：

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b} \rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。
其中该式的符号：若 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 的逆时针方向则为正，否则为负。简记为**顺负逆正**。
可以看出向量积的如下性质：

1. $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b}\rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。
其中该式的符号：若 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 的逆时针方向则为正，否则为负。简记为**顺负逆正**。
可以看出向量积的如下性质：

1. $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$
2. $|\boldsymbol{a} \times \boldsymbol{b}| = 2S_{\triangle}$

向量的向量积

两个向量的向量积，又称叉乘，其结果是一个向量，严格来说，它是仅对于三维向量的二元运算。
这个向量的模长：

$$|\boldsymbol{a} \times \boldsymbol{b}| = |\boldsymbol{a}||\boldsymbol{b}| \sin\langle \boldsymbol{a}, \boldsymbol{b} \rangle$$

这个向量的方向同时垂直于 $\boldsymbol{a}$ 和 $\boldsymbol{b}$。
那么对于二维向量，则可以简化为 $\boldsymbol{a} \times \boldsymbol{b} = x_{\boldsymbol{a}} y_{\boldsymbol{b}} - x_{\boldsymbol{b}} y_{\boldsymbol{a}}$。
其中该式的符号：若 $\boldsymbol{b}$ 在 $\boldsymbol{a}$ 的逆时针方向则为正，否则为负。简记为**顺负逆正**。
可以看出向量积的如下性质：

1. $\boldsymbol{a} \times \boldsymbol{b} = -\boldsymbol{b} \times \boldsymbol{a}$
2. $|\boldsymbol{a} \times \boldsymbol{b}| = 2S_{\triangle}$
3. $(\boldsymbol{a} \times \boldsymbol{b})^2 + (\boldsymbol{a} \cdot \boldsymbol{b})^2 = \boldsymbol{a}^2 \boldsymbol{b}^2$

向量位置关系

Problem

给定两个有共同起点的向量 $\overrightarrow{OA}$ 和 $\overrightarrow{OB}$，判断 $\overrightarrow{OA}$ 是否在 $\overrightarrow{OB}$ 的顺时针方向。

向量位置关系

Problem

给定两个有共同起点的向量 $\overrightarrow{OA}$ 和 $\overrightarrow{OB}$，判断 $\overrightarrow{OA}$ 是否在 $\overrightarrow{OB}$ 的顺时针方向。

设 $\boldsymbol{a} = \overrightarrow{OA}, \boldsymbol{b} = \overrightarrow{OB}$。

考虑作叉积，若叉积为正则 $\boldsymbol{a}$ 在 $\boldsymbol{b}$ 的顺时针方向，为负则在逆时针方向，为零则共线。

Ex. 1

判断点 $Q(x_Q, y_Q)$ 是否在线段 P_1P_2 上。

Ex. 1

判断点 $Q(x_Q, y_Q)$ 是否在线段 P_1P_2 上。

- 先判断点 Q 是否在直线 P_1P_2 上，利用叉积性质，可以用 $(Q - P_1) \times (P_2 - P_1) = 0$ 判定。

Ex. 1

判断点 $Q(x_Q, y_Q)$ 是否在线段 P_1P_2 上。

- 先判断点 Q 是否在直线 P_1P_2 上，利用叉积性质，可以用 $(Q - P_1) \times (P_2 - P_1) = 0$ 判定。
- 再判断点 Q 是否在以 P_1P_2 为对角线的矩形内。即满足：

$$\min_{i=1}^{2}(x_{P_i}) \le x_Q \le \max_{i=1}^{2}(x_{P_i}) \wedge \min_{i=1}^{2}(y_{P_i}) \le y_Q \le \max_{i=1}^{2}(y_{P_i})$$

Ex. 2

判断点 $D(x_D, y_D)$ 是否在 $\triangle ABC$ 内。

Ex. 2

判断点 $D(x_D, y_D)$ 是否在 $\triangle ABC$ 内。

法 1:

比较 $\mathrm{S} = S_{\triangle ABC}$ 和 $\mathrm{S'} = S_{\triangle ABD} + S_{\triangle BCD} + S_{\triangle ACD}$ 的大小。

若 $\mathrm{S'} = \mathrm{S}$ 则 D 在 $\triangle ABC$ 内部或其上。并且，易判点是否在三角形上，只需对于每一条边做一次 Ex. 1 的操作。

Ex. 2

判断点 $D(x_D, y_D)$ 是否在 $\triangle ABC$ 内。

法 1:
比较 S $= S_{\triangle ABC}$ 和 S' $= S_{\triangle ABD} + S_{\triangle BCD} + S_{\triangle ACD}$ 的大小。
若 S' $=$ S 则 D 在 $\triangle ABC$ 内部或其上。并且，易判点是否在三角形上，只需对于每一条边做一次 Ex. 1 的操作。
法 2：此法更偏向计算几何。根据三角形一定是凸多边形这个性质，可以简单地逆时针枚举每一条边，判断点 D 是否在每一条边的左边。
例如，判断点 D 是否在 $\overrightarrow{AB}$ 的左边，可以转化为判断 $\overrightarrow{AD}$ 是否在 $\overrightarrow{AB}$ 的逆时针方向。用叉积的正负性判断即可。

快速排斥试验

Ex. 3
判断两线段 P_1P_2 和 P_3P_4 是否相交。

快速排斥试验

Ex. 3

判断两线段 P_1P_2 和 P_3P_4 是否相交。

需要两步：快速排斥试验和跨立试验。

快速排斥试验

Ex. 3

判断两线段 P_1P_2 和 P_3P_4 是否相交。

需要两步：快速排斥试验和跨立试验。

快速排斥实验：以两条线段分别为对角线作两个矩形，判断两个矩形是否相交。

快速排斥试验

Ex. 3

判断两线段 P_1P_2 和 P_3P_4 是否相交。

需要两步：快速排斥试验和跨立试验。

快速排斥实验：以两条线段分别为对角线作两个矩形，判断两个矩形是否相交。

$$(\max(x_1,x_2) \geq \min(x_3,x_4)) \wedge (\max(x_3,x_4) \geq \min(x_1,x_2)) \wedge$$
$$(\max(y_1,y_2) \geq \min(y_3,y_4)) \wedge (\max(y_3,y_4) \geq \min(y_1,y_2))$$

跨立试验

跨立：若点 P 在直线 l 的一边且点 Q 在直线 l 的另一边，或者有点 P 或点 Q 在直线 l 上，则称线段 PQ 跨立直线 l。

跨立试验

跨立：若点 P 在直线 l 的一边且点 Q 在直线 l 的另一边，或者有点 P 或点 Q 在直线 l 上，则称线段 PQ 跨立直线 l。
跨立试验：判断每条线段是否都能跨立另一条线段所在的直线。

跨立试验

跨立：若点 P 在直线 l 的一边且点 Q 在直线 l 的另一边，或者有点 P 或点 Q 在直线 l 上，则称线段 PQ 跨立直线 l。

跨立试验：判断每条线段是否都能跨立另一条线段所在的直线。考虑怎么判定跨立。以判线段 P_3P_4 跨立直线 P_1P_2 为例，即判断向量 $\overrightarrow{P_1P_3}$ 和 $\overrightarrow{P_1P_4}$ 是否在 $\overrightarrow{P_1P_2}$ 两边。

跨立试验

跨立：若点 P 在直线 l 的一边且点 Q 在直线 l 的另一边，或者有点 P 或点 Q 在直线 l 上，则称线段 PQ 跨立直线 l。
跨立试验：判断每条线段是否都能跨立另一条线段所在的直线。
考虑怎么判定跨立。以判线段 P_3P_4 跨立直线 P_1P_2 为例，即判断向量 $\overrightarrow{P_1P_3}$ 和 $\overrightarrow{P_1P_4}$ 是否在 $\overrightarrow{P_1P_2}$ 两边。
具体实现可以用叉积，简单分类讨论整合一下，就是判断：

$$(((P_3 - P_1) \times (P_2 - P_1))((P_4 - P_1) \times (P_2 - P_1))) \leq 0 \wedge$$
$$(((P_1 - P_3) \times (P_4 - P_3))((P_2 - P_3) \times (P_4 - P_3))) \leq 0$$

Ex. 4

判断点 A 是否在多边形（点的编号按顺时针顺序依次为 $P_1, P_2, \ldots, P_n$）内。

Ex. 4

判断点 A 是否在多边形（点的编号按顺时针顺序依次为 $P_1, P_2, \ldots, P_n$）内。

首先易判断点 A 是否在多边形上。

Ex. 4

判断点 A 是否在多边形（点的编号按顺时针顺序依次为 $P_1, P_2, \ldots, P_n$）内。

首先易判断点 A 是否在多边形上。
法一：从 A 开始任意作一条射线，简单手模一下可以发现若此射线与多边形的交点个数为奇数，则点 A 在多边形内，否则点 A 在多边形外。

Ex. 4

判断点 A 是否在多边形（点的编号按顺时针顺序依次为 $P_1, P_2, \ldots, P_n$）内。

首先易判断点 A 是否在多边形上。
法一：从 A 开始任意作一条射线，简单手模一下可以发现若此射线与多边形的交点个数为奇数，则点 A 在多边形内，否则点 A 在多边形外。
注意为了防止该射线与多边形的顶点、边重合，可以使射线所在直线斜率为无理数。

Ex. 4

判断点 A 是否在多边形（点的编号按顺时针顺序依次为 $P_1, P_2, \ldots, P_n$）内。

首先易判断点 A 是否在多边形上。
法一：从 A 开始任意作一条射线，简单手模一下可以发现若此射线与多边形的交点个数为奇数，则点 A 在多边形内，否则点 A 在多边形外。
注意为了防止该射线与多边形的顶点、边重合，可以使射线所在直线斜率为无理数。
法二：计算 $\overrightarrow{AP_i}$ 和 $\overrightarrow{AP_{i+1}}$ 的有向夹角之和（$P_{n+1} = P_1$），记为 S。若有 $S = 0$ 则 A 在多边形外，否则在多边形内。

Ex. 5

判断线段 PQ 是否在多边形中。

Ex. 5
判断线段 PQ 是否在多边形中。
先判断两个端点 P 和 Q 是否在多边形中。

Ex. 5

判断线段 PQ 是否在多边形中。

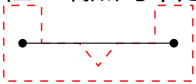
先判断两个端点 P 和 Q 是否在多边形中。
再判断线段 PQ 不与多边形的任何一条边内交。
内交：指两线段相交且两线段的任一端点均不是交点。

Ex. 5

判断线段 PQ 是否在多边形中。

先判断两个端点 P 和 Q 是否在多边形中。
再判断线段 PQ 不与多边形的任何一条边内交。
内交：指两线段相交且两线段的任一端点均不是交点。

但是会出现像这样的逆天情况：

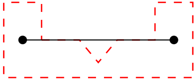


Ex. 5

判断线段 PQ 是否在多边形中。

先判断两个端点 P 和 Q 是否在多边形中。
再判断线段 PQ 不与多边形的任何一条边内交。
内交：指两线段相交且两线段的任一端点均不是交点。

但是会出现像这样的逆天情况：



所以还需要进行如下操作：
对于 P，Q 和所有多边形顶点中在线段 PQ 上的点，进行 xy 坐标排序，然后对于每对相邻点取中点判其是否在多边形内。

Ex. 6

求多边形（点的编号依次是 $P_1, P_2, \ldots, P_n$）的面积。

Ex. 6

求多边形（点的编号依次是 $P_1, P_2, \ldots, P_n$）的面积。

根据叉乘的几何意义，即 $\boldsymbol{a} \times \boldsymbol{b}$ 等于由 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 作成的平行四边形的有向面积。

Ex. 6

求多边形（点的编号依次是 $P_1, P_2, \ldots, P_n$）的面积。

根据叉乘的几何意义，即 $\boldsymbol{a} \times \boldsymbol{b}$ 等于由 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 作成的平行四边形的有向面积。具体的，若 $\boldsymbol{a}$ 逆时针转过一个小于 π 的角到 $\boldsymbol{b}$，则该面积为正，否则为负。

Ex. 6

求多边形（点的编号依次是 $P_1, P_2, \ldots, P_n$）的面积。

根据叉乘的几何意义，即 $\boldsymbol{a} \times \boldsymbol{b}$ 等于由 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 作成的平行四边形的有向面积。具体的，若 $\boldsymbol{a}$ 逆时针转过一个小于 π 的角到 $\boldsymbol{b}$，则该面积为正，否则为负。由此，设 O 为原点，我们考虑如下式子：

$$S = \frac{1}{2}\left|\sum_{i=1}^{n} \overrightarrow{OP_i}\,\overrightarrow{OP_{i+1}}\right|$$

Ex. 6

求多边形（点的编号依次是 $P_1, P_2, \ldots, P_n$）的面积。

根据叉乘的几何意义，即 $\boldsymbol{a} \times \boldsymbol{b}$ 等于由 $\boldsymbol{a}$ 和 $\boldsymbol{b}$ 作成的平行四边形的有向面积。具体的，若 $\boldsymbol{a}$ 逆时针转过一个小于 π 的角到 $\boldsymbol{b}$，则该面积为正，否则为负。由此，设 O 为原点，我们考虑如下式子：

$$S = \frac{1}{2} \left| \sum_{i=1}^{n} \overrightarrow{OP_i} \overrightarrow{OP_{i+1}} \right|$$

相当于用一个向量，起点为 O 不动，端点沿着多边形扫一圈。多边形内的区域被扫过奇数次，多边形外的区域被扫过偶数次，且正负面积互相抵消。因此，S 就是该多边形的面积。

平面凸包

对于平面上 n 个点的点集 S，覆盖该 n 个点的最小凸多边形称为平面凸包。

平面凸包

对于平面上 n 个点的点集 S，覆盖该 n 个点的最小凸多边形称为平面凸包。
计算平面凸包的算法分为两种，Andrew 算法和 Graham 扫描法。
两者时间复杂度均为 $\mathrm{O}(n \log n)$。

Graham 扫描法

首先选择一点 $P_1 \in S$ 且 P_1 是 x 值最小点，将其作为极点。

Graham 扫描法

首先选择一点 $P_1 \in S$ 且 P_1 是 x 值最小点，将其作为极点。然后将剩余点按照 $\overrightarrow{P_1 S_i}$ 的方向按顺时针排序，若方向相同则按向量模长排序，排完序后的点记为 P_i，另设目前凸包顶点序列为 V，$V_1 = P_1$。

Graham 扫描法

首先选择一点 $P_1 \in S$ 且 P_1 是 x 值最小点，将其作为极点。
然后将剩余点按照 $\overrightarrow{P_1 S_i}$ 的方向按顺时针排序，若方向相同则按向量模长排序，排完序后的点记为 P_i，另设目前凸包顶点序列为 V，$V_1 = P_1$。
当 $\overrightarrow{V_1 P_i}$ 和 $\overrightarrow{V_1 P_{i+1}}$ 方向相同时，P_i 一定不是凸包的顶点。

Graham 扫描法

首先选择一点 $P_1 \in S$ 且 P_1 是 x 值最小点，将其作为极点。然后将剩余点按照 $\overrightarrow{P_1 S_i}$ 的方向按顺时针排序，若方向相同则按向量模长排序，排完序后的点记为 P_i，另设目前凸包顶点序列为 V，$V_1 = P_1$。
当 $\overrightarrow{V_1 P_i}$ 和 $\overrightarrow{V_1 P_{i+1}}$ 方向相同时，P_i 一定不是凸包的顶点。
设目前暂定凸包顶点 t 个，考虑 P_i，若 $\overrightarrow{V_{t-1} V_t}$ 到 $\overrightarrow{V_t P_i}$ 为逆时针或 0 角度旋转时，V_{t-1}, V_t, P_i 呈凹形，V_t 一定不是凸包顶点。

Graham 扫描法

首先选择一点 $P_1 \in S$ 且 P_1 是 x 值最小点，将其作为极点。然后将剩余点按照 $\overrightarrow{P_1 S_i}$ 的方向按顺时针排序，若方向相同则按向量模长排序，排完序后的点记为 P_i，另设目前凸包顶点序列为 V，$V_1 = P_1$。
当 $\overrightarrow{V_1 P_i}$ 和 $\overrightarrow{V_1 P_{i+1}}$ 方向相同时，P_i 一定不是凸包的顶点。
设目前暂定凸包顶点 t 个，考虑 P_i，若 $\overrightarrow{V_{t-1} V_t}$ 到 $\overrightarrow{V_t P_i}$ 为逆时针或 0 角度旋转时，V_{t-1}, V_t, P_i 呈凹形，V_t 一定不是凸包顶点。
时间复杂度瓶颈在排序，为 $\mathrm{O}(n \log n)$。

算法框架

由此得到 Graham 扫描法的框架如下：

Algorithm 1: Graham's Scan

Input: $P_1, P_2, \ldots, P_n$，即初始点集

Output: $V_1, V_2, \ldots, V_m$，即凸包顶点点集

$V_1 \leftarrow P_1$;// P_1 是指 x 值最小的点

$j \leftarrow 1$;

按照 $\overrightarrow{V_1 P_j}$ 的旋转方向排序 P_2 到 P_n;

for $i \leftarrow 2$ **to** n **do**

  while $(i < n) \wedge (\overrightarrow{V_1 P_i} \times \overrightarrow{V_1 P_{i+1}} = 0)$ **do**

    $i \leftarrow i + 1$;

  while $(j > 1) \wedge (\overrightarrow{V_{j-1} V_j} \times \overrightarrow{V_j P_i} \geq 0)$ **do**

    $j \leftarrow j - 1$;

  $j \leftarrow j + 1$;

  $V_j \leftarrow P_i$;

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。
显然排序后最前和最后的元素一定在凸包上。

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。
显然排序后最前和最后的元素一定在凸包上。
并且，根据凸包的性质，若从任一凸包上的顶点往逆时针方向走，轨迹总是左拐的。

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。
显然排序后最前和最后的元素一定在凸包上。
并且，根据凸包的性质，若从任一凸包上的顶点往逆时针方向走，轨迹总是左拐的。
因此，可以用单调栈维护，一旦出现右拐则说明不在凸包上。

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。
显然排序后最前和最后的元素一定在凸包上。
并且，根据凸包的性质，若从任一凸包上的顶点往逆时针方向走，轨迹总是左拐的。
因此，可以用单调栈维护，一旦出现右拐则说明不在凸包上。
由于是按坐标排序，遍历一次并不能找出整个凸包，所以要正着遍历一次找出上凸壳，倒着遍历一次找出下凸壳。

Andrew 算法

首先将所有点以 x 坐标为第一关键字，y 坐标为第二关键字排序。
显然排序后最前和最后的元素一定在凸包上。
并且，根据凸包的性质，若从任一凸包上的顶点往逆时针方向走，轨迹总是左拐的。
因此，可以用单调栈维护，一旦出现右拐则说明不在凸包上。
由于是按坐标排序，遍历一次并不能找出整个凸包，所以要正着遍历一次找出上凸壳，倒着遍历一次找出下凸壳。
时间复杂度瓶颈同样在是排序的复杂度。

P3194 [HNOI2008] 水平可见直线

在平面上有互不重合的 $n(n \leq 5 \times 10^5)$ 条直线 $L_1, L_2, \ldots L_n$，若在 $y = +\infty$ 处往下看，能见到 L_i 的某个子线段，则称 L_i 为可见的，求出所有可见的直线。

P3194 [HNOI2008] 水平可见直线

在平面上有互不重合的 $n(n \leq 5 \times 10^5)$ 条直线 $L_1, L_2, \ldots L_n$，若在 $y = +\infty$ 处往下看，能见到 L_i 的某个子线段，则称 L_i 为可见的，求出所有可见的直线。

首先，这些可见的子线段一定组成一个下凸壳，其斜率是从小变大的。

P3194 [HNOI2008] 水平可见直线

在平面上有互不重合的 $n(n \leq 5 \times 10^5)$ 条直线 $L_1, L_2, \ldots L_n$，若在 $y = +\infty$ 处往下看，能见到 L_i 的某个子线段，则称 L_i 为可见的，求出所有可见的直线。

首先，这些可见的子线段一定组成一个下凸壳，其斜率是从小变大的。

因此可以将直线按斜率排序，斜率相同的直线只需算截距最大的即可。

P3194 [HNOI2008] 水平可见直线

在平面上有互不重合的 $n(n \le 5 \times 10^5)$ 条直线 $L_1, L_2, \ldots L_n$，若在 $y = +\infty$ 处往下看，能见到 L_i 的某个子线段，则称 L_i 为可见的，求出所有可见的直线。

首先，这些可见的子线段一定组成一个下凸壳，其斜率是从小变大的。

因此可以将直线按斜率排序，斜率相同的直线只需算截距最大的即可。

维护一个单调栈，当前栈顶的直线若被完全覆盖则弹出，做完了。

Atcoder typical90_ao

给你 n 个放置在整点上的钉子，问至少加上多少钉子使得在外面围一圈绳子，其包含的所有整点均有钉子。

Atcoder typical90_ao

给你 n 个放置在整点上的钉子，问至少加上多少钉子使得在外面围一圈绳子，其包含的所有整点均有钉子。

这是 N_z_大师搬的题。

https://atcoder.jp/contests/typical90/tasks/typical90_ao

Atcoder typical90_ao

给你 n 个放置在整点上的钉子，问至少加上多少钉子使得在外面围一圈绳子，其包含的所有整点均有钉子。

这是 N_z_大师搬的题。

https://atcoder.jp/contests/typical90/tasks/typical90_ao

首先，这个绳子围成的区域一定是这些点组成的凸包，从而转化为计算凸包内及凸包上的整点个数，再减去 n 即可。

Atcoder typical90_ao

给你 n 个放置在整点上的钉子，问至少加上多少钉子使得在外面围一圈绳子，其包含的所有整点均有钉子。

这是 N_z_大师搬的题。
https://atcoder.jp/contests/typical90/tasks/typical90_ao
首先，这个绳子围成的区域一定是这些点组成的凸包，从而转化为计算凸包内及凸包上的整点个数，再减去 n 即可。
考虑 Pick 公式：对于一个多边形，其面积为 S，多边形内整点个数为 p，多边形上整点个数为 q，则一定有 $S = p + \frac{1}{2}q - 1$。

对于本题，可以用叉积计算出 S。

对于本题，可以用叉积计算出 S。
对于两端点都在整点上的线段 AB，线段上的整点个数为 $\gcd(|x_A - x_B|, |y_A - y_B|) + 1$。

对于本题，可以用叉积计算出 S。
对于两端点都在整点上的线段 AB，线段上的整点个数为 $\gcd(|x_A - x_B|, |y_A - y_B|) + 1$。
所以整个多边形上的整点个数为 $q = \sum \gcd(|x_{P_i} - x_{P_{i+1}}|, |y_{P_i} - y_{P_{i+1}}|)$。
逆用 Pick 公式求出 p 即可。

切线和对踵点对

对于一个凸多边形 P，**切线** l 是一条与 P 相交且 P 的内部在 l 的一侧的线。

切线和对踵点对

对于一个凸多边形 P，**切线** l 是一条与 P 相交且 P 的内部在 l 的一侧的线。
如果属于 P 的两个点 p, q 在凸多边形 P 的两条平行切线上，则称 p, q 是一对**对踵点对**。

切线和对踵点对

对于一个凸多边形 P，**切线** l 是一条与 P 相交且 P 的内部在 l 的一侧的线。
如果属于 P 的两个点 p, q 在凸多边形 P 的两条平行切线上，则称 p, q 是一对**对踵点对**。
注意：两条平行切线确定了至少一对对踵点对。具体的，有“点-点”，“点-边”，“边-边”对踵点对。

切线和对踵点对

对于一个凸多边形 P，**切线** l 是一条与 P 相交且 P 的内部在 l 的一侧的线。
如果属于 P 的两个点 p, q 在凸多边形 P 的两条平行切线上，则称 p, q 是一对**对踵点对**。
注意：两条平行切线确定了至少一对对踵点对。具体的，有“点-点”，“点-边”，“边-边”对踵点对。

- 当两条平行切线与凸多边形只有两个交点时，出现“点-点”。
- 当其中一条切线与凸多边形的一条边共线时，出现“点-边”。
- 当两条平行切线与凸多边形的两条边共线时，出现“边-边”。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。
事实上，直径是由多边形的平行切线的最远距离决定的。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。
事实上，直径是由多边形的平行切线的最远距离决定的。
考虑对于任意一条边，找到距离该边最远的点。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。
事实上，直径是由多边形的平行切线的最远距离决定的。
考虑对于任意一条边，找到距离该边最远的点。
显然这样做是 $O(n^2)$ 的。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。
事实上，直径是由多边形的平行切线的最远距离决定的。
考虑对于任意一条边，找到距离该边最远的点。
显然这样做是 $\mathrm{O}(n^2)$ 的。
但是由于凸多边形的凸性，当逆时针遍历多边形的边时，其对应的最远的点同样也是按逆时针旋转的。

凸多边形的直径

直径：凸多边形上任意两点间的距离的最大值。
事实上，直径是由多边形的平行切线的最远距离决定的。
考虑对于任意一条边，找到距离该边最远的点。
显然这样做是 $\mathrm{O}(n^2)$ 的。
但是由于凸多边形的凸性，当逆时针遍历多边形的边时，其对应的最远的点同样也是按逆时针旋转的。
这样做就可以达到 $\mathrm{O}(n)$ 的复杂度。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

1. 计算多边形 y 最小值所在点 $y_{\min}$ 和 y 最大值所在点 $y_{\max}$。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

1. 计算多边形 y 最小值所在点 $y_{\min}$ 和 y 最大值所在点 $y_{\max}$。
2. 通过 $y_{\min}$ 和 $y_{\max}$ 构造两条水平切线。计算两点之间的距离，维护为当前距离最小值。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

1. 计算多边形 y 最小值所在点 $y_{\min}$ 和 y 最大值所在点 $y_{\max}$。
2. 通过 $y_{\min}$ 和 $y_{\max}$ 构造两条水平切线。计算两点之间的距离，维护为当前距离最小值。
3. 同时旋转两条线直到其中一条与多边形的一条边重合。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

1. 计算多边形 y 最小值所在点 $y_{\min}$ 和 y 最大值所在点 $y_{\max}$。
2. 通过 $y_{\min}$ 和 $y_{\max}$ 构造两条水平切线。计算两点之间的距离，维护为当前距离最小值。
3. 同时旋转两条线直到其中一条与多边形的一条边重合。
4. 一对新的对踵点对此时产生，计算距离，更新当前最小值。

凸多边形的宽度

凸多边形的**宽度**定义为平行切线间最小距离。
显然，若一对平行切线都不与多边形的任意一条边重合则总能通过旋转减少它们之间的距离。
这就意味着只需要考虑“点-边”对踵点对和“边-边”对踵点对。
过程如下：

1. 计算多边形 y 最小值所在点 $y_{\min}$ 和 y 最大值所在点 $y_{\max}$。
2. 通过 $y_{\min}$ 和 $y_{\max}$ 构造两条水平切线。计算两点之间的距离，维护为当前距离最小值。
3. 同时旋转两条线直到其中一条与多边形的一条边重合。
4. 一对新的对踵点对此时产生，计算距离，更新当前最小值。
5. 重复步骤 3 和 4，直到产生的对踵点对为与最初的平行切线重合时结束。

凸多边形间的最大距离

两个凸多边形 P, Q 的**最大距离**是指 $\max_{p \in P, q \in Q} \{\mathrm{dis}_{p,q}\}$。

凸多边形间的最大距离

两个凸多边形 P, Q 的**最大距离**是指 $\max\limits_{p\in P, q\in Q}\{\mathrm{dis}_{p,q}\}$。
另定义两个凸多边形间的对踵点对 p, q，类似于凸多边形的对踵点对，只是平行切线是有向且反向的。

凸多边形间的最大距离

两个凸多边形 P, Q 的**最大距离**是指 $\max\limits_{p\in P, q\in Q}\{\mathrm{dis}_{p,q}\}$。
另定义两个凸多边形间的对踵点对 p, q，类似于凸多边形的对踵点对，只是平行切线是有向且反向的。
则凸多边形间的最大距离由多边形间对踵点对决定。

凸多边形间的最大距离

两个凸多边形 P, Q 的**最大距离**是指 $\max\limits_{p\in P, q\in Q}\{\mathrm{dis}_{p,q}\}$。

另定义两个凸多边形间的对踵点对 p, q，类似于凸多边形的对踵点对，只是平行切线是有向且反向的。

则凸多边形间的最大距离由多边形间对踵点对决定。

因此，可以用类似于上一问的方法，找出所有对踵点对计算距离。

凸多边形间的最大距离

两个凸多边形 P, Q 的**最大距离**是指 $\max\limits_{p\in P, q\in Q}\{\mathrm{dis}_{p,q}\}$。
另定义两个凸多边形间的对踵点对 p, q，类似于凸多边形的对踵点对，只是平行切线是有向且反向的。
则凸多边形间的最大距离由多边形间对踵点对决定。
因此，可以用类似于上一问的方法，找出所有对踵点对计算距离。
同理可以求出凸多边形间最小距离，不多赘述。

凸多边形最小面积外接矩形

Luogu P3187

给定一个凸多边形 P，求出其最小面积外接矩形。

凸多边形最小面积外接矩形

Luogu P3187

给定一个凸多边形 P，求出其最小面积外接矩形。

依然可以使用旋转卡壳法来解决这个问题。

凸多边形最小面积外接矩形

Luogu P3187

给定一个凸多边形 P，求出其最小面积外接矩形。

依然可以使用旋转卡壳法来解决这个问题。
显然，这个外接矩形一定有至少一条边与多边形的边重合，因此考虑遍历每一条多边形的边。

凸多边形最小面积外接矩形

Luogu P3187

给定一个凸多边形 P，求出其最小面积外接矩形。

依然可以使用旋转卡壳法来解决这个问题。
显然，这个外接矩形一定有至少一条边与多边形的边重合，因此考虑遍历每一条多边形的边。
利用叉积维护距离其最远的点，利用点积维护最左、最右的点，计算面积即可。

POJ2079 Triangle

给定 $n(3 \leq n \leq 5 \times 10^4)$ 个整点的点集，且坐标的绝对值小于等于 10^4。求选出三个点组成的三角形面积最大值。保证存在非退化三角形。

POJ2079 Triangle

给定 $n(3 \le n \le 5 \times 10^4)$ 个整点的点集，且坐标的绝对值小于等于 10^4。求选出三个点组成的三角形面积最大值。保证存在非退化三角形。

考虑求出这个点集的纯净凸包，即凸包中须排除连续三点共线。

POJ2079 Triangle

给定 $n(3 \leq n \leq 5 \times 10^4)$ 个整点的点集，且坐标的绝对值小于等于 10^4。求选出三个点组成的三角形面积最大值。保证存在非退化三角形。

考虑求出这个点集的纯净凸包，即凸包中须排除连续三点共线。则很显然的，这三个点一定在这个凸包上。

POJ2079 Triangle

给定 $n(3 \leq n \leq 5 \times 10^4)$ 个整点的点集，且坐标的绝对值小于等于 10^4。求选出三个点组成的三角形面积最大值。保证存在非退化三角形。

考虑求出这个点集的纯净凸包，即凸包中须排除连续三点共线。
则很显然的，这三个点一定在这个凸包上。
考虑暴力旋转卡壳。按一定顺序枚举两个点，那么另外一个点有单调性，则时间复杂度是 $\mathrm{O}(n \log n + m^2)$，$m$ 是凸包上的点的数量。

POJ2079 Triangle

给定 $n(3 \leq n \leq 5 \times 10^4)$ 个整点的点集，且坐标的绝对值小于等于 10^4。求选出三个点组成的三角形面积最大值。保证存在非退化三角形。

考虑求出这个点集的纯净凸包，即凸包中须排除连续三点共线。则很显然的，这三个点一定在这个凸包上。
考虑暴力旋转卡壳。按一定顺序枚举两个点，那么另外一个点有单调性，则时间复杂度是 $\mathrm{O}(n \log n + m^2)$，$m$ 是凸包上的点的数量。
这样做复杂度其实是对的，因为有三个限制：点是整点，凸包无连续三点共线，坐标绝对值小于 10^4，设 $N = 2 \times 10^4$，那么这样凸包上点数是 $O(N^{\frac{2}{3}})$ 级别的，即小于 800 个点。
这个结论的证明需涉及大量数论。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。
洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。
对点集 S，求出其凸包，设 S' 是在凸包内的点集，继续对点集 S' 做如上操作直到 S' 为空。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。

洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。
对点集 S，求出其凸包，设 S' 是在凸包内的点集，继续对点集 S' 做如上操作直到 S' 为空。
这样做完后，会留下层层嵌套的凸包。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。

洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。
对点集 S，求出其凸包，设 S' 是在凸包内的点集，继续对点集 S' 做如上操作直到 S' 为空。
这样做完后，会留下层层嵌套的凸包。
两个嵌套的凸包间的区域称为**环面**。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。

洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。
对点集 S，求出其凸包，设 S' 是在凸包内的点集，继续对点集 S' 做如上操作直到 S' 为空。
这样做完后，会留下层层嵌套的凸包。
两个嵌套的凸包间的区域称为**环面**。
可以利用旋转卡壳在线性时间内求出环面的三角剖分。具体地，可以对于两个凸包分别作出位于 x 坐标最小的铅垂切线，然后同方向旋转、连边即可。

洋葱三角剖分

给定一个平面上的点集，目标是构造一个点集上的三角剖分。

洋葱三角剖分是一种基于对点集“剥洋葱皮”的操作。
对点集 S，求出其凸包，设 S' 是在凸包内的点集，继续对点集 S' 做如上操作直到 S' 为空。
这样做完后，会留下层层嵌套的凸包。
两个嵌套的凸包间的区域称为**环面**。
可以利用旋转卡壳在线性时间内求出环面的三角剖分。具体地，可以对于两个凸包分别作出位于 x 坐标最小的铅垂切线，然后同方向旋转、连边即可。
洋葱三角剖分形成的图是哈密顿图。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

1. 取点集中 x 坐标最小点 A，并过点 A 作铅垂线。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

1. 取点集中 x 坐标最小点 A，并过点 A 作铅垂线。
2. 按照顺时针转动直线，直到直线碰到另一个点集中的点，将两点连线。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

1. 取点集中 x 坐标最小点 A，并过点 A 作铅垂线。
2. 按照顺时针转动直线，直到直线碰到另一个点集中的点，将两点连线。
3. 旋转中心变为在新碰到的点，并继续重复步骤 2，但是总忽略已经碰到过的点。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

1. 取点集中 x 坐标最小点 A，并过点 A 作铅垂线。
2. 按照顺时针转动直线，直到直线碰到另一个点集中的点，将两点连线。
3. 旋转中心变为在新碰到的点，并继续重复步骤 2，但是总忽略已经碰到过的点。

实际上，一个点集的凸螺旋线和“洋葱皮”可以在线性时间内相互转化。

螺旋三角剖分

点集的螺旋三角剖分是基于集合螺旋凸包的三角剖分图。
凸螺旋线可以通过如下方法构造：

1. 取点集中 x 坐标最小点 A，并过点 A 作铅垂线。
2. 按照顺时针转动直线，直到直线碰到另一个点集中的点，将两点连线。
3. 旋转中心变为在新碰到的点，并继续重复步骤 2，但是总忽略已经碰到过的点。

实际上，一个点集的凸螺旋线和“洋葱皮”可以在线性时间内相互转化。
对于一个凸螺旋线的三角剖分，实际上是基于环面三角剖分，但却要更为复杂，需要将其分割为合适的凸包链。

合并凸包

给定两个凸多边形，求出包含他们并的最小凸多边形。

合并凸包

给定两个凸多边形，求出包含他们并的最小凸多边形。
此即合并凸包。

合并凸包

给定两个凸多边形，求出包含他们并的最小凸多边形。
此即合并凸包。
显然有一种做法是对于多边形上的所有点求一次凸包。

合并凸包

给定两个凸多边形，求出包含他们并的最小凸多边形。
此即合并凸包。
显然有一种做法是对于多边形上的所有点求一次凸包。
更高效的方法是依赖于查找多边形间的桥。

合并凸包

给定两个凸多边形，求出包含他们并的最小凸多边形。

此即合并凸包。
显然有一种做法是对于多边形上的所有点求一次凸包。
更高效的方法是依赖于查找多边形间的桥。
对于多边形 $P = \{p_1, p_2, \dots, p_n\}, Q = \{q_1, q_2, \dots, q_n\}$，点对 (p_i, q_j) 形成 P 和 Q 之间的桥当且仅当：
(p_i, q_j) 形成一对对踵点对，且 $p_{i-1}, p_{i+1}, q_{i-1}, q_{i+1}$ 都位于 $p_i q_j$ 这条线的同一侧。

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

1. 分别计算 P 和 Q 拥有最大 y 坐标（若有多个则取最大 x 坐标）的点。

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

1. 分别计算 P 和 Q 拥有最大 y 坐标（若有多个则取最大 x 坐标）的点。
2. 构造经过这两个点的水平切线，并且切线均指向 x 轴正方向。

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

1. 分别计算 P 和 Q 拥有最大 y 坐标（若有多个则取最大 x 坐标）的点。
2. 构造经过这两个点的水平切线，并且切线均指向 x 轴正方向。
3. 同时顺时针旋转两条切线直到其中一条与边相交，此时产生至少一对新的对踵点对，判断其是否能形成桥。

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

1. 分别计算 P 和 Q 拥有最大 y 坐标（若有多个则取最大 x 坐标）的点。
2. 构造经过这两个点的水平切线，并且切线均指向 x 轴正方向。
3. 同时顺时针旋转两条切线直到其中一条与边相交，此时产生至少一对新的对踵点对，判断其是否能形成桥。
4. 持续旋转直到切线回到原来的位置。

假设多边形的点是以顺时针顺序给定的，则 $\mathrm{O}(n)$ 算法如下：

1. 分别计算 P 和 Q 拥有最大 y 坐标（若有多个则取最大 x 坐标）的点。
2. 构造经过这两个点的水平切线，并且切线均指向 x 轴正方向。
3. 同时顺时针旋转两条切线直到其中一条与边相交，此时产生至少一对新的对踵点对，判断其是否能形成桥。
4. 持续旋转直到切线回到原来的位置。
5. 所有可能的桥已经确定，通过用凸包链连续连接桥来构造合并凸包。

临界切线

两个凸多边形间的**临界切线**（又称 CS 线）是使得两个多边形分居线不同侧的切线。CS 线只会在两个顶点处与两个多边形相交。

临界切线

两个凸多边形间的**临界切线**（又称 CS 线）是使得两个多边形分居线不同侧的切线。CS 线只会在两个顶点处与两个多边形相交。因此，一条 CS 线由多边形间顶点对 (p_i, q_j) 确定，该点对满足：(p_i, q_j) 是一对对踵点对且 p_{i-1}, p_{i+1} 位于线 $p_i q_j$ 一侧，同时 $q_{j-1}\,q_{j+1}$ 位于另一侧。

临界切线

两个凸多边形间的**临界切线**（又称 CS 线）是使得两个多边形分居线不同侧的切线。CS 线只会在两个顶点处与两个多边形相交。因此，一条 CS 线由多边形间顶点对 (p_i, q_j) 确定，该点对满足：(p_i, q_j) 是一对对踵点对且 p_{i-1}, p_{i+1} 位于线 $p_i q_j$ 一侧，同时 $q_{j-1}\, q_{j+1}$ 位于另一侧。
利用这个结论，CS 线可以很容易地确定，只需要检查多边形间的对踵点对。

凸包属性

要计算对踵点对那么就用旋转卡壳。

要计算对踵点对那么就用旋转卡壳。

1. 计算 P 上 y 坐标值最小的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\max}Q$）。

要计算对踵点对那么就用旋转卡壳。

1. 计算 P 上 y 坐标值最小的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\max}Q$）。
2. 在 $y_{\min}P$ 和 $y_{\max}Q$ 处构造两条切线 l_P 和 l_Q 使得它们对应的多边形对于它们的右侧。此时 l_P 和 l_Q 是异向的，且 $y_{\min}P$ 和 $y_{\max}Q$ 成为了一对对踵点对，判定其是否能成为 CS 线。

要计算对踵点对那么就用旋转卡壳。

1. 计算 P 上 y 坐标值最小的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\max}Q$）。
2. 在 $y_{\min}P$ 和 $y_{\max}Q$ 处构造两条切线 l_P 和 l_Q 使得它们对应的多边形对于它们的右侧。此时 l_P 和 l_Q 是异向的，且 $y_{\min}P$ 和 $y_{\max}Q$ 成为了一对对踵点对，判定其是否能成为 CS 线。
3. 旋转这两条边，知道其中一条和其对应的多边形的边重合，此时产生至少一对对踵点对，判定其是否能成为 CS 线。

要计算对踵点对那么就用旋转卡壳。

1. 计算 P 上 y 坐标值最小的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\min}P$）和 Q 上 y 坐标值最大的顶点（称为 $y_{\max}Q$）。
2. 在 $y_{\min}P$ 和 $y_{\max}Q$ 处构造两条切线 l_P 和 l_Q 使得它们对应的多边形对于它们的右侧。此时 l_P 和 l_Q 是异向的，且 $y_{\min}P$ 和 $y_{\max}Q$ 成为了一对对踵点对，判定其是否能成为 CS 线。
3. 旋转这两条边，知道其中一条和其对应的多边形的边重合，此时产生至少一对对踵点对，判定其是否能成为 CS 线。
4. 重复第 3 步，直到产生的点对是 $(y_{\min}P, y_{\max}Q)$。

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。
这东西有个更常见的名称：闵可夫斯基和。其具有如下性质：

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。
这东西有个更常见的名称：闵可夫斯基和。其具有如下性质：

1. $P+Q$ 是凸多边形。

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。
这东西有个更常见的名称：闵可夫斯基和。其具有如下性质：

1. $P+Q$ 是凸多边形。
2. 顶点集 $P+Q$ 是顶点集 P 和 Q 的和。

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。
这东西有个更常见的名称：闵可夫斯基和。其具有如下性质：

1. $P+Q$ 是凸多边形。
2. 顶点集 $P+Q$ 是顶点集 P 和 Q 的和。
3. 顶点集 $P+Q$ 是 P 和 Q 间的并踵点对集。

凸多边形矢量和

给定平面上两个凸多边形 P 和 Q，定义 P 和 Q 的矢量和记为 $P+Q=\{p+q|p\in P\wedge q\in Q\}$。
这东西有个更常见的名称：闵可夫斯基和。其具有如下性质：

1. $P+Q$ 是凸多边形。
2. 顶点集 $P+Q$ 是顶点集 P 和 Q 的和。
3. 顶点集 $P+Q$ 是 P 和 Q 间的并踵点对集。
4. 设 P 有 m 个顶点，Q 有 n 个顶点，则 $P+Q$ 有不多于 $m+n$ 个顶点。

凸包属性

下面的结论提供了一个 $O(n)$ 用增量法求矢量和的方法。

凸包属性

下面的结论提供了一个 $O(n)$ 用增量法求矢量和的方法。
设 $P+Q$ 的向量为 $z_k = p_i + q_j$，在 p_i 和 q_j 处构造两条平行切线，并且使得多边形均位于各自线的右侧。

下面的结论提供了一个 $O(n)$ 用增量法求矢量和的方法。
设 $P+Q$ 的向量为 $z_k = p_i + q_j$，在 p_i 和 q_j 处构造两条平行切线，并且使得多边形均位于各自线的右侧。
则两条平行切线和 $p_i p_{i+1}$ 会形成夹角，分别记为 θ_i, ϕ_j。

下面的结论提供了一个 $O(n)$ 用增量法求矢量和的方法。
设 $P+Q$ 的向量为 $z_k = p_i + q_j$，在 p_i 和 q_j 处构造两条平行切线，并且使得多边形均位于各自线的右侧。
则两条平行切线和 p_ip_{i+1} 会形成夹角，分别记为 θ_i, ϕ_j。

因此，下一个向量 $z_{k+1} = \begin{cases} p_{i+1} + q_j & \theta_i < \phi_j \\ p_i + q_{j+1} & \theta_i > \phi_j \\ p_{i+1} + q_{j+1} & \theta_i = \phi_j \end{cases}$

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax + by + c \geq 0$ 或 $ax + by + c \leq 0$ 确定。

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax + by + c \geq 0$ 或 $ax + by + c \leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax+by+c\geq 0$ 或 $ax+by+c\leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。
对于若干个半平面的交集，即满足所有半平面的点的点集，称为**半平面交**。

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax + by + c \geq 0$ 或 $ax + by + c \leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。
对于若干个半平面的交集，即满足所有半平面的点的点集，称为**半平面交**。
半平面和半平面交有如下性质：

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax+by+c\geq 0$ 或 $ax+by+c\leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。
对于若干个半平面的交集，即满足所有半平面的点的点集，称为**半平面交**。
半平面和半平面交有如下性质：

1. 若该半平面或半平面交无界，则添加四个半平面保证其有界，此时它是个凸多边形区域。

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax + by + c \geq 0$ 或 $ax + by + c \leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。
对于若干个半平面的交集，即满足所有半平面的点的点集，称为**半平面交**。
半平面和半平面交有如下性质：

1. 若该半平面或半平面交无界，则添加四个半平面保证其有界，此时它是个凸多边形区域。
2. n 个半平面的交 $\bigcap\limits_{i=1}^{n} H_i$ 是一个至多 n 条边的凸多边形。

半平面交

在平面直角坐标系中，**半平面**是平面上的一条直线及其一侧的总和，可由 $ax+by+c\geq 0$ 或 $ax+by+c\leq 0$ 确定。
在 OI 中，半平面常以向量的一边（均为右边或均为左边）给定。
对于若干个半平面的交集，即满足所有半平面的点的点集，称为**半平面交**。
半平面和半平面交有如下性质：

1. 若该半平面或半平面交无界，则添加四个半平面保证其有界，此时它是个凸多边形区域。
2. n 个半平面的交 $\bigcap_{i=1}^{n} H_i$ 是一个至多 n 条边的凸多边形。

实际上，半平面交后的区域，有可能是一个直线、射线、线段、点甚至空集。

两个凸多边形的交

给定两个凸多边形 P 和 Q，求它们的交。

两个凸多边形的交

给定两个凸多边形 P 和 Q，求它们的交。
发现没怎么讲扫描线，稍微涉及一下。

两个凸多边形的交

给定两个凸多边形 P 和 Q，求它们的交。

发现没怎么讲扫描线，稍微涉及一下。

假设我们有条铅垂的扫描线，从左向右扫描，任意时刻它与两个凸多边形的交点一定不超过 4 个。

两个凸多边形的交

给定两个凸多边形 P 和 Q，求它们的交。

发现没怎么讲扫描线，稍微涉及一下。
假设我们有条铅垂的扫描线，从左向右扫描，任意时刻它与两个凸多边形的交点一定不超过 4 个。
那么就可以通过这至多四个交点去计算出交的多边形的内部区域。

两个凸多边形的交

给定两个凸多边形 P 和 Q，求它们的交。

发现没怎么讲扫描线，稍微涉及一下。
假设我们有条铅垂的扫描线，从左向右扫描，任意时刻它与两个凸多边形的交点一定不超过 4 个。
那么就可以通过这至多四个交点去计算出交的多边形的内部区域。
显然我们不能对于所有有理数扫描，因此应扫描所有点的 x 坐标。时间复杂度 $\mathrm{O}(n)$。

求 n 个半平面的交

求 n 个半平面的交

这里讲两种做法，D&C 算法和 S&I 算法。

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。
其实分治很好想，分为如下步骤：

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。
其实分治很好想，分为如下步骤：

1. 将 n 个半平面分成两个 $\frac{n}{2}$ 的集合；

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。
其实分治很好想，分为如下步骤：

1. 将 n 个半平面分成两个 $\frac{n}{2}$ 的集合；
2. 对子集进行递归求解。

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。
其实分治很好想，分为如下步骤：

1. 将 n 个半平面分成两个 $\frac{n}{2}$ 的集合；
2. 对子集进行递归求解。
3. 然后用刚刚的求两个凸多边形的交的算法进行合并。

D&C 算法

D&C 算法，即 Divide and Conquer Algorithm，基于分治思想。
其实分治很好想，分为如下步骤：

1. 将 n 个半平面分成两个 $\frac{n}{2}$ 的集合；
2. 对子集进行递归求解。
3. 然后用刚刚的求两个凸多边形的交的算法进行合并。

时间复杂度 $\mathrm{O}(n\log n)$。

S&I 算法

S&I 算法，即 Sort and Incremental Algorithm，排序增量算法。

S&I 算法

S&I 算法，即 Sort and Incremental Algorithm，排序增量算法。定义半平面的**极角**为其对应向量和 x 轴所形成的夹角。可以用 $\operatorname{atan2}(y, x)$ 求。

S&I 算法

S&I 算法，即 Sort and Incremental Algorithm，排序增量算法。
定义半平面的**极角**为其对应向量和 x 轴所形成的夹角。可以用 $\operatorname{atan2}(y, x)$ 求。
以下半平面均视为在对应向量的左侧。

1. 将半平面按照极角排序，若极角相同则越靠左排在越前面。

1. 将半平面按照极角排序，若极角相同则越靠左排在越前面。
2. 维护一个单调双端队列。若新加进来的半平面在队列尾端两个半平面的交点的左边，则进行 pop_back。若新加进来的半平面在队列前的两个半平面的交点的左边，则进行 pop_front，最后 push_back 该半平面，并计算其与队列尾半平面的交点。**注意**：先 pop_back 再 pop_front。

1. 将半平面按照极角排序，若极角相同则越靠左排在越前面。
2. 维护一个单调双端队列。若新加进来的半平面在队列尾端两个半平面的交点的左边，则进行 pop_back。若新加进来的半平面在队列前的两个半平面的交点的左边，则进行 pop_front，最后 push_back 该半平面，并计算其与队列尾半平面的交点。**注意**：先 pop_back 再 pop_front。
3. 考虑最后是形成一个凸多边形，即类似环，因此还需要将队列首的半平面当作新加的半平面再去 pop_back，去除多余半平面。

1. 将半平面按照极角排序，若极角相同则越靠左排在越前面。
2. 维护一个单调双端队列。若新加进来的半平面在队列尾端两个半平面的交点的左边，则进行 pop_back。若新加进来的半平面在队列前的两个半平面的交点的左边，则进行 pop_front，最后 push_back 该半平面，并计算其与队列尾半平面的交点。**注意**：先 pop_back 再 pop_front。
3. 考虑最后是形成一个凸多边形，即类似环，因此还需要将队列首的半平面当作新加的半平面再去 pop_back，去除多余半平面。
4. 还在队列中的相邻两个半平面的交点连起来就是半平面交的结果了。

1. 将半平面按照极角排序，若极角相同则越靠左排在越前面。
2. 维护一个单调双端队列。若新加进来的半平面在队列尾端两个半平面的交点的左边，则进行 pop_back。若新加进来的半平面在队列前的两个半平面的交点的左边，则进行 pop_front，最后 push_back 该半平面，并计算其与队列尾半平面的交点。**注意**：先 pop_back 再 pop_front。
3. 考虑最后是形成一个凸多边形，即类似环，因此还需要将队列首的半平面当作新加的半平面再去 pop_back，去除多余半平面。
4. 还在队列中的相邻两个半平面的交点连起来就是半平面交的结果了。

时间复杂度瓶颈在第一步，为 $O(n \log n)$。

FAQ

小学就学过的一些东西

- 向量
- 基础算法

平面凸包

旋转卡壳

- 计算距离
- 三角剖分
- 凸包属性

半平面交

杂题

P6247 [SDOI2012] 最近最远点对

给定一个 n 个点的点集，求其中任意两点的最短距离及最长距离。

最长距离就是旋转卡壳板子题，主要考虑怎么求最短距离。

最长距离就是旋转卡壳板子题，主要考虑怎么求最短距离。
按 x 坐标对点从左到右排序，然后考虑分治。

最长距离就是旋转卡壳板子题，主要考虑怎么求最短距离。
按 x 坐标对点从左到右排序，然后考虑分治。
子区间处理完后要合并信息，即要求计算在 $[l, \mathrm{mid}]$ 和 $[\mathrm{mid}+1, r]$ 两边各取一点的最短距离。

最长距离就是旋转卡壳板子题，主要考虑怎么求最短距离。
按 x 坐标对点从左到右排序，然后考虑分治。
子区间处理完后要合并信息，即要求计算在 $[l, \mathrm{mid}]$ 和 $[\mathrm{mid}+1, r]$ 两边各取一点的最短距离。
设 $d = \min(ans_{[l,\mathrm{mid}]}, ans_{[\mathrm{mid}+1,r]})$，则可以将点集缩小为 $\{p_i \mid |x_i - x_{\mathrm{mid}}| < d\}$。

最长距离就是旋转卡壳板子题，主要考虑怎么求最短距离。
按 x 坐标对点从左到右排序，然后考虑分治。
子区间处理完后要合并信息，即要求计算在 $[l, \mathrm{mid}]$ 和 $[\mathrm{mid}+1, r]$ 两边各取一点的最短距离。
设 $d = \min(ans_{[l,\mathrm{mid}]}, ans_{[\mathrm{mid}+1,r]})$，则可以将点集缩小为 $\{p_i \mid |x_i - x_{\mathrm{mid}}| < d\}$。
在这点集里按 y 坐标排序，然后暴力枚举 y 坐标差小于 d 两点计算距离。由于对于任意点只会与至多 5 个点计算距离，所以复杂度是正确的。
总复杂度是 $\mathrm{O}(n \log n)$。

P3299 [SDOI2013] 保护出题人

共 n 关植物大战僵尸，第 i 个僵尸 z_i 的生命为 a_i。第 i 关依次有 $z_i, z_{i-1}, z_{i-2}, \ldots, z_1$ 共 i 个僵尸，相邻僵尸相距 d，首个僵尸距离植物 x_i，所有僵尸速度均为 $1/s$。你需要钦定植物在每关的攻击力 y_i/s，使得 $\sum y_i$ 最小并输出之。

首先，若将僵尸的生命值加上其前面僵尸的生命值，并且植物的攻击同时作用于所有僵尸。发现这并不影响答案。

首先，若将僵尸的生命值加上其前面僵尸的生命值，并且植物的攻击同时作用于所有僵尸。发现这并不影响答案。考虑朴素做法，设 $s_i = \sum_{j=1}^{i} a_j$，则对于任意 i 的答案是：

$$\max_{j=1}^{i} \left\{ \frac{s_i - s_{j-1}}{x_i + d(i-j)} \right\}$$

首先，若将僵尸的生命值加上其前面僵尸的生命值，并且植物的攻击同时作用于所有僵尸。发现这并不影响答案。考虑朴素做法，设 $s_i = \sum_{j=1}^{i} a_j$，则对于任意 i 的答案是：

$$\max_{j=1}^{i}\left\{\frac{s_i - s_{j-1}}{x_i + d(i-j)}\right\}$$

发现这是一个斜率的形式，即 (dj, s_{j-1}) 和 $(x_i + di, s_i)$ 形成直线的斜率最大。发现全都是定点。

首先，若将僵尸的生命值加上其前面僵尸的生命值，并且植物的攻击同时作用于所有僵尸。发现这并不影响答案。考虑朴素做法，设 $s_i = \sum_{j=1}^{i} a_j$，则对于任意 i 的答案是：

$$\max_{j=1}^{i} \left\{ \frac{s_i - s_{j-1}}{x_i + d(i-j)} \right\}$$

发现这是一个斜率的形式，即 (dj, s_{j-1}) 和 $(x_i + di, s_i)$ 形成直线的斜率最大。发现全都是定点。
因此可以对前者维护一个下凸包，在上面二分出最大斜率即可。

P3829 [SHOI2012] 信用卡凸包

将 n 个完全相同的圆角矩形分别以给定角度摆放在平面上的给定位置，求其凸包周长。$n \leq 10^4$。

显然有一个结论是凸包的每个转角处都是一段圆弧，所有圆弧加起来刚好是一个圆。

显然有一个结论是凸包的每个转角处都是一段圆弧，所有圆弧加起来刚好是一个圆。
将所有圆角矩形的圆心做一遍凸包。然后将凸包上的边可以平移至原圆角矩形上，发现边与边之间只差了一段圆弧。

显然有一个结论是凸包的每个转角处都是一段圆弧，所有圆弧加起来刚好是一个圆。
将所有圆角矩形的圆心做一遍凸包。然后将凸包上的边可以平移至原圆角矩形上，发现边与边之间只差了一段圆弧。
因此答案就是圆心的凸包周长加圆的周长。

P2924 [USACO08DEC]Largest Fence G

给定 $n(n \leq 250)$ 个点（不存在三点共线），问能够选出的最多数量的点使得能够组成一个凸多边形。

凸多边形存在这样一个性质：存在一个顶点 P，使得从 P 开始一直往一个方向走，边的极角是单调的。

凸多边形存在这样一个性质：存在一个顶点 P，使得从 P 开始一直往一个方向走，边的极角是单调的。
因此可以 n^2 暴力两两连边，把所有边按极角序排序。

凸多边形存在这样一个性质：存在一个顶点 P，使得从 P 开始一直往一个方向走，边的极角是单调的。
因此可以 n^2 暴力两两连边，把所有边按极角序排序。
考虑 dp，枚举起始点 P，然后按极角序遍历边做 dp 就可以了。
时间复杂度 $\mathrm{O}(n^3)$。

UVA1303 Wall

给定 n 个点，求最短的包含所有点的轮廓且任何点到轮廓的距离均不小于 L。$n \leq 10^3$。

先对这 n 个点求出凸包。

先对这 n 个点求出凸包。
距离不小于 L，那么就尽量让它变成距离等于 L。那么就可以以凸包上的顶点为圆心作半径为 L 的圆。

先对这 n 个点求出凸包。
距离不小于 L，那么就尽量让它变成距离等于 L。那么就可以以凸包上的顶点为圆心作半径为 L 的圆。
显然这个轮廓不能与圆有交，则转化为对这些东西求凸包。用信用卡凸包一题的结论就做完了。答案即凸包周长加圆的周长。

P3222 [HNOI2012] 射箭

给定 $n(n \leq 10^5)$ 条在第一象限中的平行于 y 轴的线段，第 i 条线段端点坐标为 $(x_i, y_{1,i})$ 和 $(x_i, y_{2,i})$（$y_{1,i} < y_{2,i}$）。求最大的 k 使得存在一条经过原点、开口向下的抛物线满足其与编号 $1 - k$ 的线段都有交点。

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。
设抛物线方程 $y = ax^2 + bx$，若其与线段 i 有交点，则有不等式方程：

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。
设抛物线方程 $y = ax^2 + bx$，若其与线段 i 有交点，则有不等式方程：

$$y_{1,i} \le ax_i^2 + bx_i \le y_{2,i}$$

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。
设抛物线方程 $y = ax^2 + bx$，若其与线段 i 有交点，则有不等式方程：

$$y_{1,i} \le ax_i^2 + bx_i \le y_{2,i}$$

$$\frac{y_{1,i}}{x_i} \le ax_i + b \le \frac{y_{2,i}}{x_i}$$

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。
设抛物线方程 $y = ax^2 + bx$，若其与线段 i 有交点，则有不等式方程：

$$y_{1,i} \le ax_i^2 + bx_i \le y_{2,i}$$

$$\frac{y_{1,i}}{x_i} \le ax_i + b \le \frac{y_{2,i}}{x_i}$$

可以分成两个半平面 $ax_i + b - \frac{y_{1,i}}{x_i} \ge 0$ 和 $ax_i + b - \frac{y_{2,i}}{x_i} \le 0$。

二分答案，设当前二分至 d，考虑如何判定 d 是否合法。
设抛物线方程 $y = ax^2 + bx$，若其与线段 i 有交点，则有不等式方程：

$$y_{1,i} \leq ax_i^2 + bx_i \leq y_{2,i}$$

$$\frac{y_{1,i}}{x_i} \leq ax_i + b \leq \frac{y_{2,i}}{x_i}$$

可以分成两个半平面 $ax_i + b - \frac{y_{1,i}}{x_i} \geq 0$ 和 $ax_i + b - \frac{y_{2,i}}{x_i} \leq 0$。
于是对所有 $2d$ 个半平面做半平面交，最后判交集是否为空即可。

P3297 [SDOI2013] 逃考

在一个 $(0,0)-(x_l,y_l)$ 的矩形中，小杨小小杨在 (x_0,y_0) 上。矩形中另有 $n(n \leq 600)$ 个亲戚，矩形中的任意一点被距离其最近的亲戚所控制。

小杨想要走到矩形的边上，求他最少走过多少个亲戚的控制区域。

考虑如何求出亲戚 i 的控制区域。

考虑如何求出亲戚 i 的控制区域。
作出他与其他亲戚的中垂线，将这些中垂线向自己一边的半平面作一次半平面交，得到的交集即为控制区域。事实上，这样生成的多边形叫做泰森多边形。

考虑如何求出亲戚 i 的控制区域。
作出他与其他亲戚的中垂线，将这些中垂线向自己一边的半平面作一次半平面交，得到的交集即为控制区域。事实上，这样生成的多边形叫做泰森多边形。
这样转化成了一个图论问题，将控制区域有公共边的亲戚（或者亲戚和矩形边）连边，求一次最短路即可。

P4557 [JSOI2018] 战争

A, B 是平面上的点集，分别有 n，m 个点。定义一点 P 被一点集 S 控制当且仅当在 S 中存在三点，它们组成的三角形（可退化）包含点 P（含边界）。若存在一点被两个点集 A, B 共同控制，那么称 A, B 存在冲突。

q 次询问，每次询问给定一个向量，将 B 点集都按向量平移至 B'，问 A 和 B' 是否冲突。$1 \le n, m, q \le 10^5$

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。

设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：

$b + \boldsymbol{w} = a$

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。
设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：

$$b + \boldsymbol{w} = a$$

$$\boldsymbol{w} = a - b$$

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。
设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：

$$b + \boldsymbol{w} = a$$

$$\boldsymbol{w} = a - b$$

$$\boldsymbol{w} \in \{a - b | a \in A \wedge b \in B\} = A + (-B)$$

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。

设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：

$b + \boldsymbol{w} = a$

$\boldsymbol{w} = a - b$

$\boldsymbol{w} \in \{a - b | a \in A \wedge b \in B\} = A + (-B)$

发现这东西就是闵可夫斯基和。只需要将 B 的 x, y 坐标都取反即可。

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。
设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：
$b + \boldsymbol{w} = a$
$\boldsymbol{w} = a - b$
$\boldsymbol{w} \in \{a - b | a \in A \wedge b \in B\} = A + (-B)$
发现这东西就是闵可夫斯基和。只需要将 B 的 x, y 坐标都取反即可。
问题转化为判断一点是否在一个凸包内。

对于一个点集，其控制区域实际上是它的凸包。冲突即两个凸包存在交点。

设询问的向量是 $\boldsymbol{w}$，假设存在一点 $b \in B$，平移后在点 $a \in A$，那么有：

$b + \boldsymbol{w} = a$

$\boldsymbol{w} = a - b$

$\boldsymbol{w} \in \{a - b | a \in A \wedge b \in B\} = A + (-B)$

发现这东西就是闵可夫斯基和。只需要将 B 的 x, y 坐标都取反即可。

问题转化为判断一点是否在一个凸包内。

可以在左半凸包和右半凸包分别二分，找出 $y = x_w$ 经过的凸包上的两条边，判断该点是否在两边之间。时间复杂度 $O(n \log n)$。

P3291 [SCOI2016] 妖怪

给定 n 个点 (x_i, y_i)，记 $f_i = x_i + y_i + x_i k + \frac{y_i}{k}$，其中 k 是任意正实数，求 $\max\{f_i\}$ 的最小值。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。
它的横纵截距分别为 $x_i + \frac{y_i}{k}$ 和 $y_i + kx_i$。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。
它的横纵截距分别为 $x_i + \frac{y_i}{k}$ 和 $y_i + kx_i$。
发现要求的 f_i 刚好是这样一条直线的横纵截距之和。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。
它的横纵截距分别为 $x_i + \frac{y_i}{k}$ 和 $y_i + kx_i$。
发现要求的 f_i 刚好是这样一条直线的横纵截距之和。
由于斜率已经钦定是 $-k$，因此当直线与这 n 个点的上凸包相切时 f_i 取得最大值。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。
它的横纵截距分别为 $x_i + \frac{y_i}{k}$ 和 $y_i + kx_i$。
发现要求的 f_i 刚好是这样一条直线的横纵截距之和。
由于斜率已经钦定是 $-k$，因此当直线与这 n 个点的上凸包相切时 f_i 取得最大值。
逆向思考，对于上凸包上的某一点 p_x，当 f_x 在某一 k 是所有 f 中最大值时，所形成的直线一定与上凸包相切于 p_x。

考虑这样一条直线，斜率为 $-k$ 且经过 (x_i, y_i)，则直线表达式为 $y = -kx + y_i + kx_i$。
它的横纵截距分别为 $x_i + \frac{y_i}{k}$ 和 $y_i + kx_i$。
发现要求的 f_i 刚好是这样一条直线的横纵截距之和。
由于斜率已经钦定是 $-k$，因此当直线与这 n 个点的上凸包相切时 f_i 取得最大值。
逆向思考，对于上凸包上的某一点 p_x，当 f_x 在某一 k 是所有 f 中最大值时，所形成的直线一定与上凸包相切于 p_x。
记 $p_{x-1}p_x$ 斜率为 k_1，p_xp_{x+1} 斜率为 k_2，则合法斜率区间在 $[k_2, k_1]$。问题转化为在该范围内求 f_x 最小值。

根据均值不等式，理想状态下的 f_x 最小值是当 k 取 $\sqrt{\frac{y_i}{x_i}}$。

根据均值不等式，理想状态下的 f_x 最小值是当 k 取 $\sqrt{\frac{y_i}{x_i}}$。

若 $-\sqrt{\frac{y_i}{x_i}}$ 在上述斜率区间则已经求出了 f_x 的最小值。

根据均值不等式，理想状态下的 f_x 最小值是当 k 取 $\sqrt{\frac{y_i}{x_i}}$。

若 $-\sqrt{\frac{y_i}{x_i}}$ 在上述斜率区间则已经求出了 f_x 的最小值。

若不在合法斜率区间内，考虑原 f_x 实际上是个双勾函数，越贴近最小值点函数值越小。因此检查区间端点处的函数值即可。

时间复杂度瓶颈在求凸包，为 $O(n \log n)$。