

E5-B题

20374249 刘益洲





题面

B 旋转卡壳怎么读

时间限制：1000ms 内存限制：65536kb

通过率：141/162 (87.04%) 正确率：141/1013 (13.92%)

题目描述

教练闫发现二维平面上有 n 个点，坐标分别为 $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ ，定义两点 (x_i, y_i) 和 (x_j, y_j) 的欧几里得距离为 $\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ 。教练闫要求同学们求出任意两点间最大欧几里得距离，即：

$$\max_{1 \leq i < j \leq n} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

输入格式

第一行一个正整数 n ($2 \leq n \leq 10^5$)，表示点的数量。

接下来 n 行，每行两个至多六位小数的浮点数 x_i, y_i ($0 \leq |x_i|, |y_i| \leq 5 \times 10^5$)，表示第 i 个点的坐标。

输出格式

输出一行一个浮点数，保留六位小数，表示任意两点间最大欧几里得距离。

输入样例

```
4
0 1
1 0
1 1
-1 -1
```

输出样例

2.828427



分析

平面最远点对

或者说凸多边形直径

暴力枚举 $O(n^2)$

$$2 \leq n \leq 10^5$$



分析

平面最远点对

或者说凸多边形直径

暴力枚举



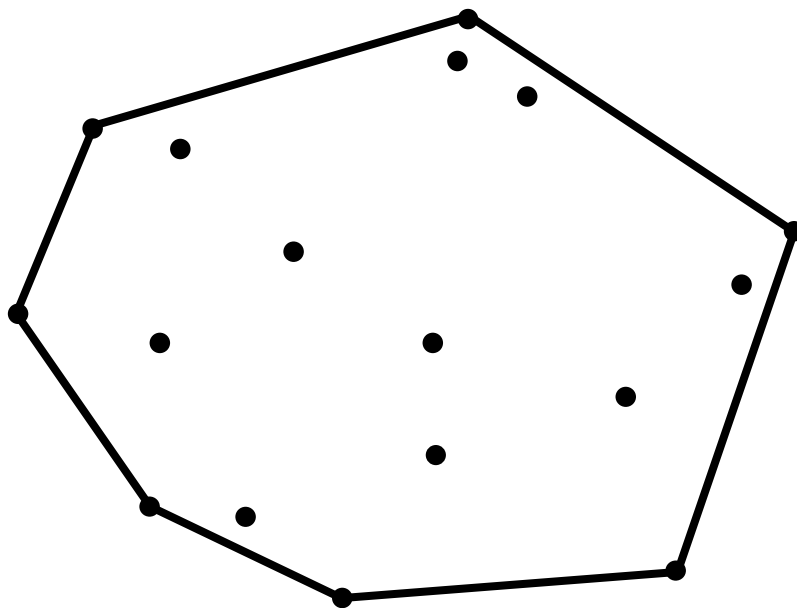
打咩 zenzen打咩



分析

凸包

注意到平面最远点对一定出现在凸包上，简单证明见下：



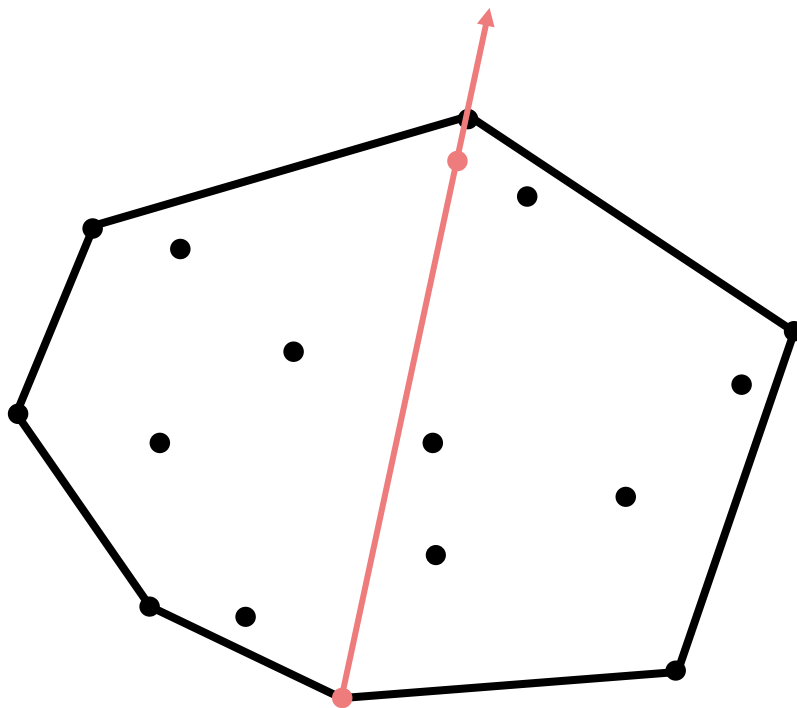


分析

凸包

注意到平面最远点对一定出现在凸包上，简单证明见下：

假设最远点对中有一个点不在凸包上，连线，假如经过凸包上某点，则显然这不是最远点对。

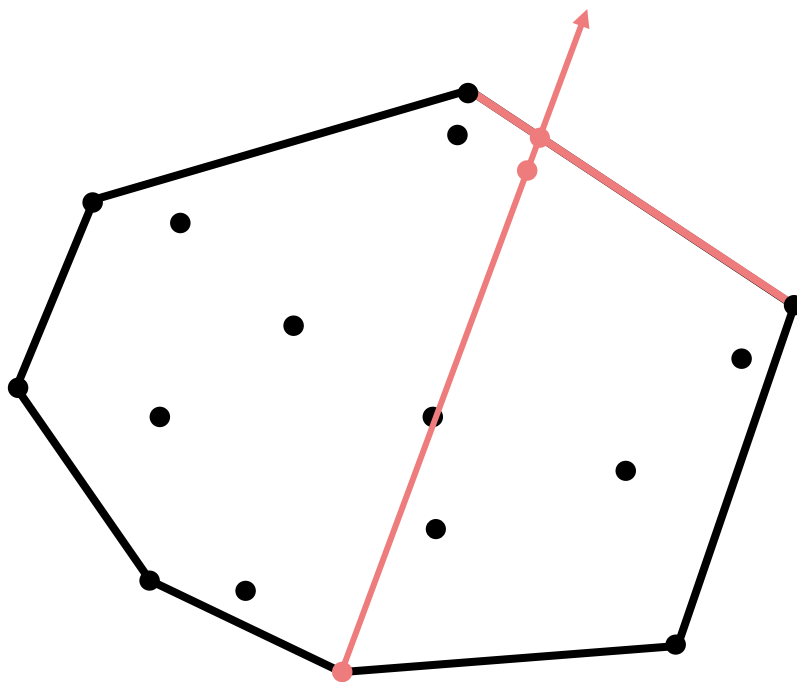




分析

凸包

注意到平面最远点对一定出现在凸包上，简单证明见下：
否则所连直线将经过凸包某边。



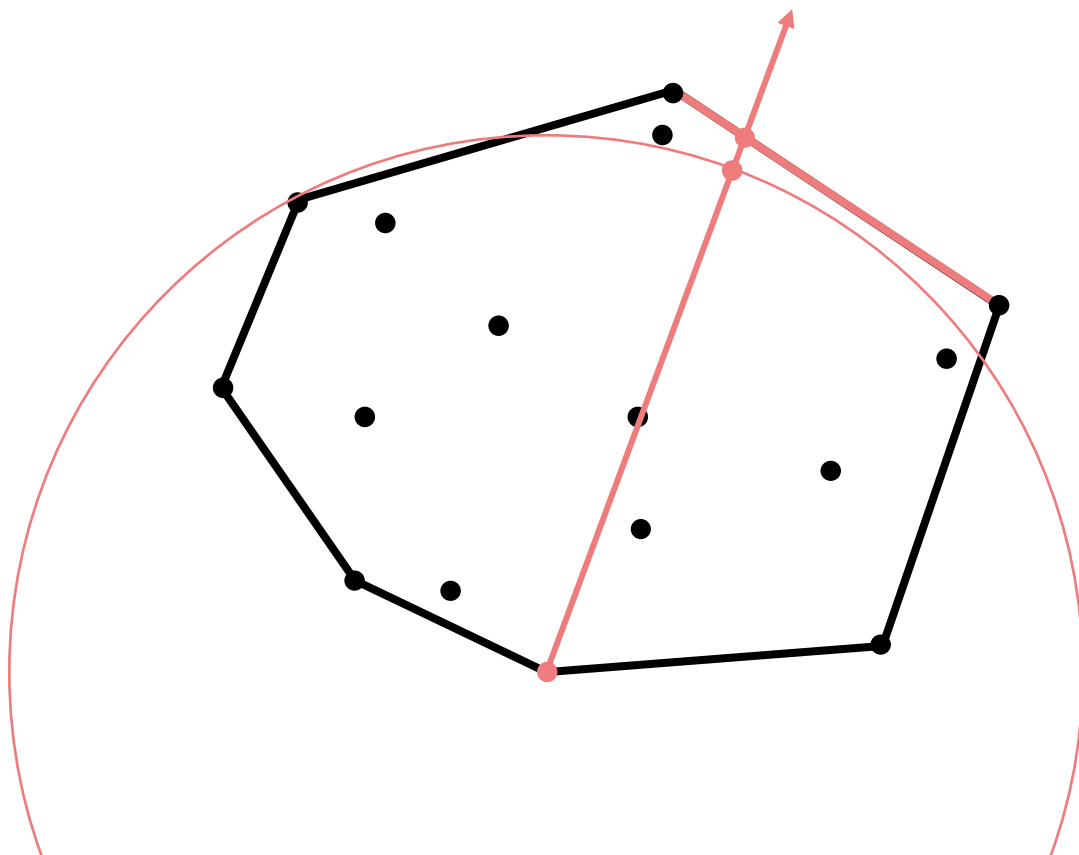


凸包

分析

注意到平面最远点对一定出现在凸包上，简单证明见下：

此时这条过园外一点的线段，其两端点不可能都在园内，与平面最远点对矛盾。

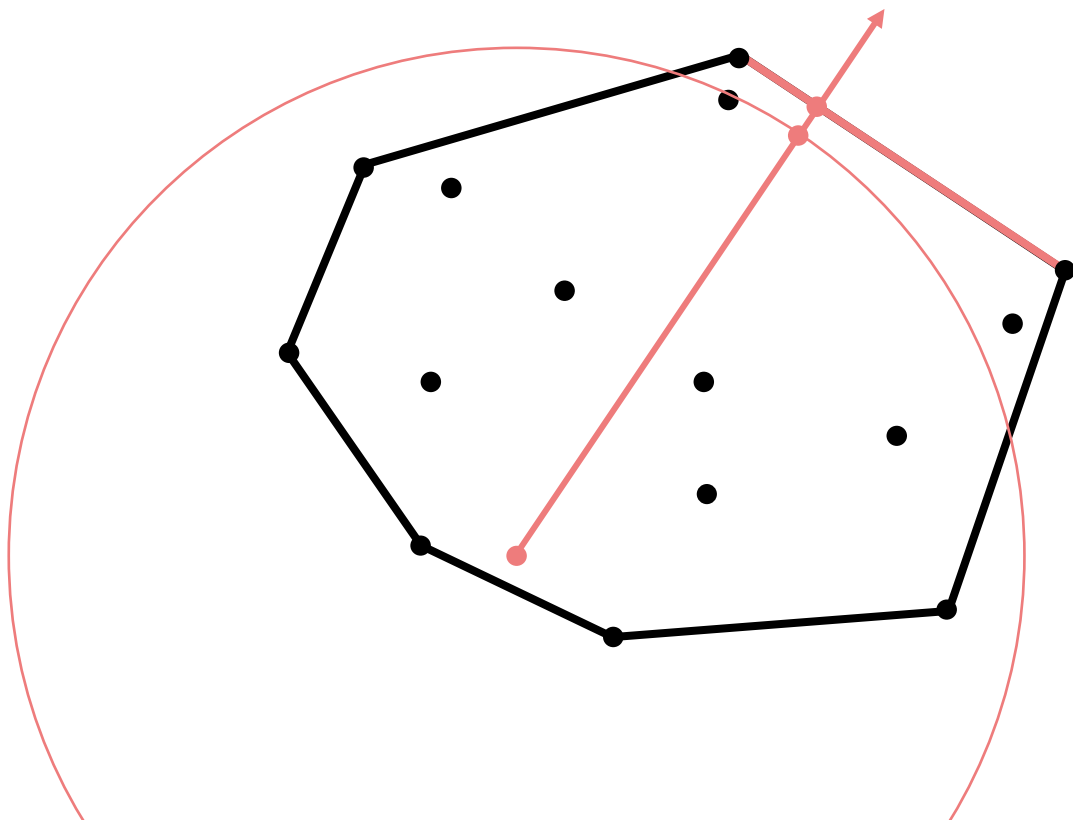




分析

凸包

注意到平面最远点对一定出现在凸包上，简单证明见下：
两点都不在凸包上同理。

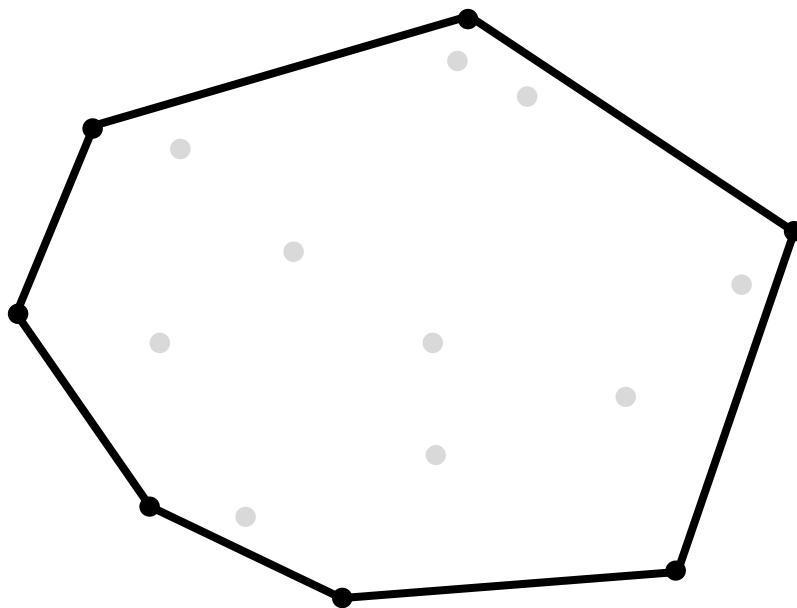




分析

凸包

使用Graham扫描法，使用合适的排序算法，可以在 $O(n \log_2 n)$ 的时间内求解得到图形的凸包。

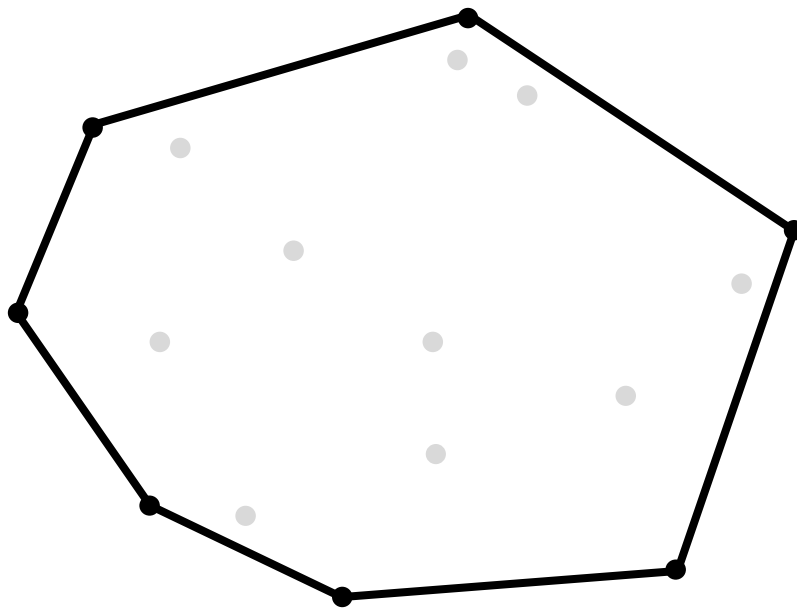




分析

旋转卡壳

使用Graham扫描法，使用合适的排序算法，可以在 $O(n \log_2 n)$ 的时间内求解得到图形的凸包。接下来我们求解最远点对。



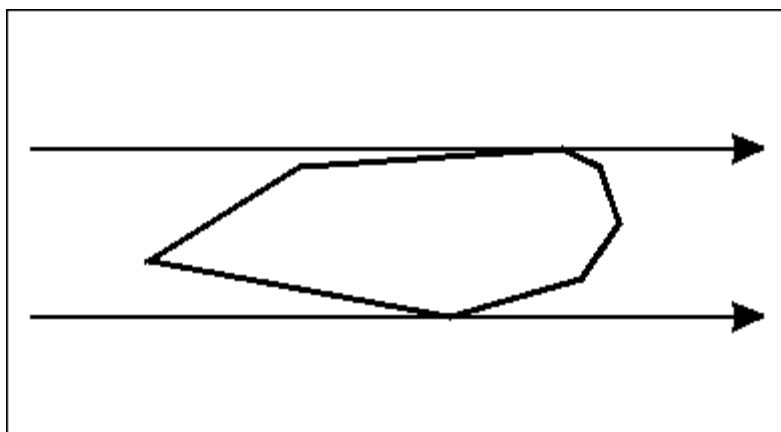


分析

旋转卡壳

如果过凸多边形上两点作一对平行线，使得整个多边形都在这两条线之间，那么这两个点被称为一对对踵点。

直径一定会在对踵点中产生，如果两个点不是对踵点，那么两个点中一定可以让一个点向另一个点的对踵点方向移动使得距离更大。并且点与点之间的距离可以体现为线与线之间的距离，在非对踵点之间构造平行线，一定没有在对踵点构造平行线优，这一点可以通过平移看出。

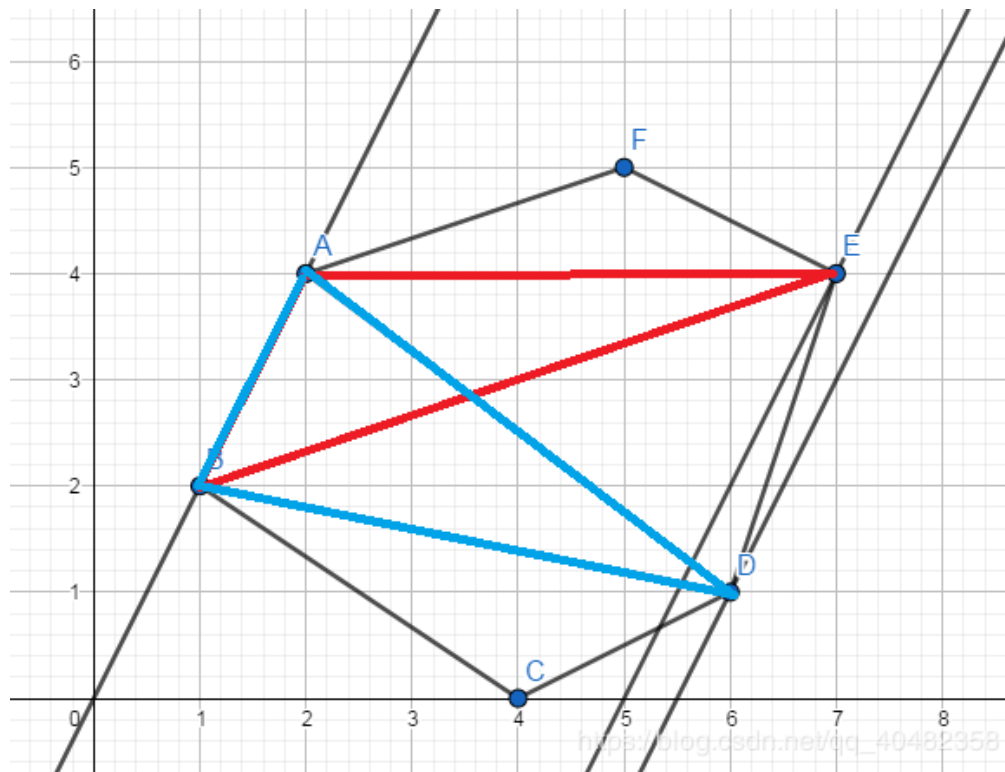




分析

旋转卡壳

通过旋转的方式寻找对踵点较为复杂，同时精度较低，于是采用面积计算距离得到对踵点。

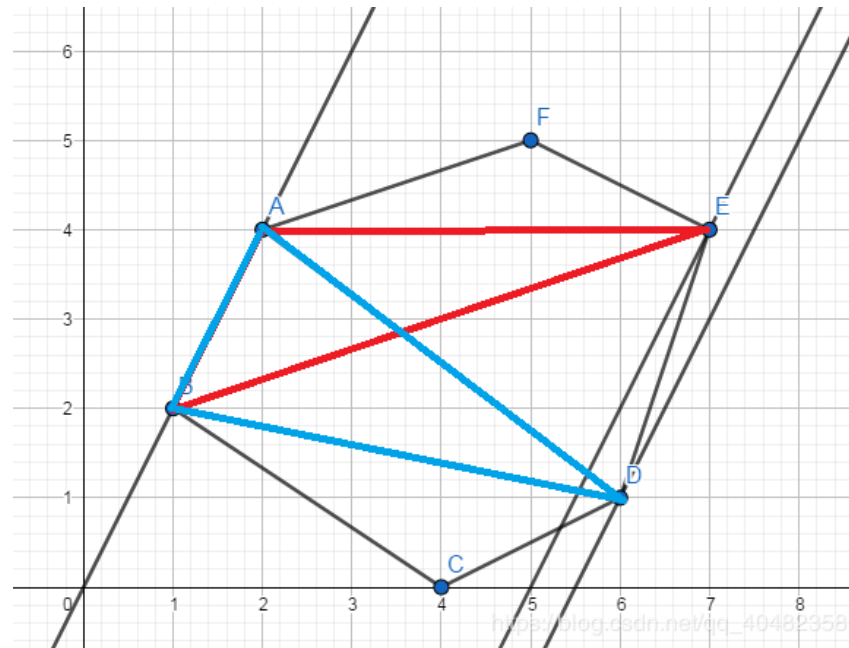




分析

旋转卡壳

大体的思路就是通过叉积计算面积，使用面积的大小代表距离的大小，来寻找距离当前边最远的点，即得到两对对踵点。然后从中选取距离较长的一对，并于当前记录的最大值。

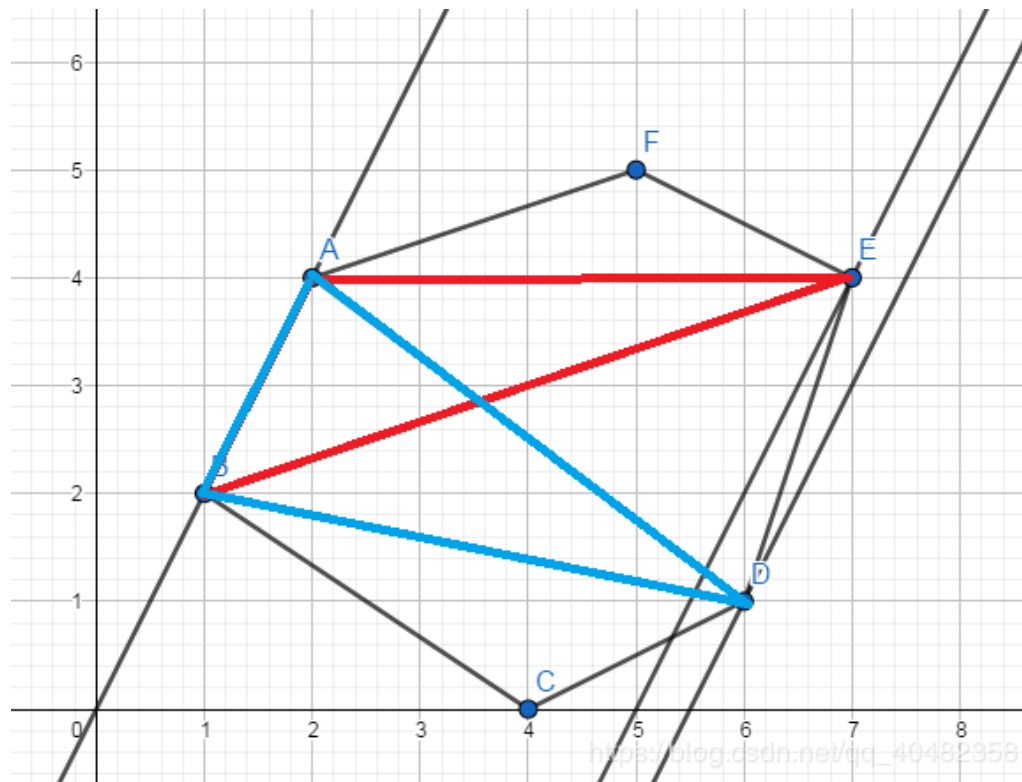




分析

旋转卡壳

容易发现，在找寻对踵点对时仅逆时针旋转，不会发生向前搜索的情况，故该算法具有 $O(n)$ 的时间复杂度。





代码

旋转卡壳

关键代码见下，注意将第一个元素添加至末尾，并对当前搜寻到的点取模。

```
double rotatingCalipersConvexDiameter(vector<Point> A)
{
    int now = 0, n = A.size();
    double ans = 0;
    A.push_back(A[0]);
    for (int i = 0; i < n; i++)
    {
        while (((A[i + 1] - A[i]) ^ (A[now] - A[i])) < ((A[i + 1] - A[i]) ^ (A[now + 1] - A[i]))) now = (now + 1) % n;
        ans = max(ans, max(A[i].dis(A[now]), A[i + 1].dis(A[now])));
    }
    return ans;
}
```




划水

旋转卡壳怎么读

我们假设有 n 个字，每个字最多有 m 种读音

暴力枚举



打咩 zenzen打咩



划水

旋转卡壳怎么读

我们假设有 n 个字，每个字最多有 m 种读音

暴力枚举 $O(m^n)$

$$n = 4, m = 2$$

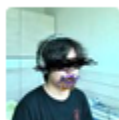
对于本题，这个上界很紧



划水

旋转卡壳怎么读

众说纷纭



周子寒

no more 高三狗

19 人赞同了该回答

旋 (xuàn) 转 (zhuǎn) 卡 (kǎ) 壳 (ké)

计算几何-旋转卡壳

读法

—(其实我也不知道该怎么读，有16种读法)—

xuàn zhuàn qiǎ qiào

旋转卡壳

xuán zhuǎn kǎ ké

旋转卡壳

Bai du 汉语

旋转

旋转

[xuán zhuǎn]

知乎 @zzd233

Bai du 汉语

卡壳

卡壳

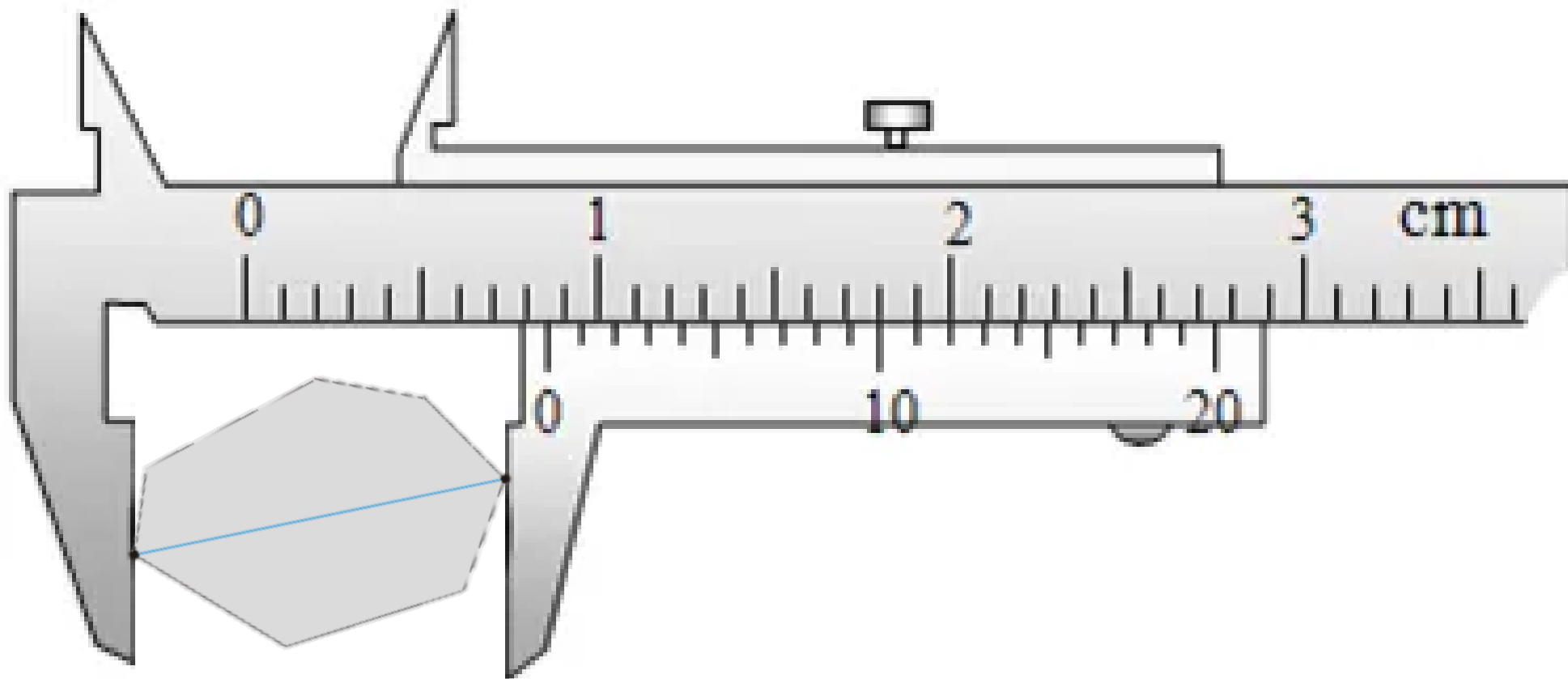
[qiǎ ké]



划水

旋转卡壳怎么读

我的建议是读旋转卡尺



谢谢大家

20374249 刘益洲

