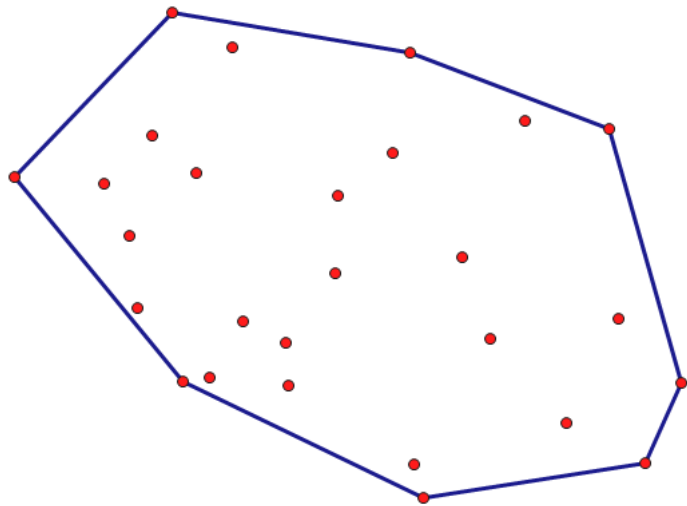


凸包--Graham扫描法

一直听大佬们说：凸包、凸包、凸包
一直不会。。。。。
然后。。。
今天考试，考了一道计算几何的简单题。。。
这，，，还是学一下吧。。
然后考试现场学习一下凸包算法。

先理解一下凸包是啥东西。



http://blog.csdn.net/qq_30974369

看看这张图
解释一下凸包是什么
如果你有一堆点（原谅我画的很凌乱）
那么，找到一个点集
依次连接这些点
使他们形成一个凸多边形
并且所有的点都包括在这个多边形的内部或者边上
这个多边形就是一个凸包（我写的肯定一点也不严谨）

不管怎么样，就先这样理解一下吧。。。。。

凸包是啥应该不难理解，那么，给你一堆点，怎么求凸包？

这种东西。。。。。
先大概说一下把。。

首先找到最靠近左下的那个点，这个点一定在凸包上（不难理解吧。。画个图就知道了）

以这个点为极点，其他点按照极角排序

然后按照顺序依次访问所有点，判断可行性

这样子干说真是虚无缥缈的东西。。。。。
画图来解释

公告

AmazingCounters.com

给我写信

About Me

坐标：HN-CS-CJ
已经退役的大菜鸡。蒟蒻yyb的
QQ:1357828232
喂喂喂，加我QQ的验证问题填yyb就好啦啦
QwQ
请备注一下年级和学校还有您的名字（缩写就行啦）
欢迎大家来交换友链

有史以来最菜的人

垃圾yyb的CSDN博客

同一届的巨佬们

萝卜
zzzzsy
YCB
the_Despair!
NeosKnight
Cyhlhj pp!!!!(AFO)
fdfdf(AFO)
FlashHu(AFO)
lalaxu!!xzy!!(AFO)
mona!(AFO)
zctoylm 小胖(AFO)
Tyher(AFO)
eternal风度 伊(AFO)
ysn(AFO)
dwq(AFO)
Cwen(AFO)
Brioche lkj(AFO)
特殊部分：单向orz 大聚聚cx233666(怎么大聚聚也AFO了啊喂)

学长们

wfj_2048(AFO)
贱狗老师(AFO)

外校的大佬们

Redbag(ylx)
XSC肖啾啾
Bill Yang
dkw!
xMinh
YYJciji(有人要求写"麓山第一巨佬"?)
(被称作麓山第二的?) jjeff小蒟蒻
Refun?Aufun!!!
苏卿念
poorpool破池姐姐!
神仙yww
litble!!!!
zip-shadow又吊打我了
Mychael!
zhouzhendong
zyk
fwat
ErkkiErkko
子谦。
slr
Dispwnl
Little_Jian
Paulliant
CDQZ dxy
memset0
wjyy
Qiuly
CDSS ldx
PhantasmDragon
Tgotp
lk!!!
xht37
heanda
zgjjj
sigongzi
筱柒_littleseven

将(tian)来(tian)爆踩我们的学弟们---

M-sea
YCH, smy
鸡贼贼、呆鸡、屎鸡
杜杜熊、杜老师
球球
猫贼贼、屎猫
糖姐姐ij
切题无数的ltst tq
heyujun
xxz
Qihoo360
gj尻
hbx

昵称：小蒟蒻yyb
园龄：3年

粉丝： 323

关注： 36

+加关注

< 2019年10月 >						
日	一	二	三	四	五	六
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

搜索

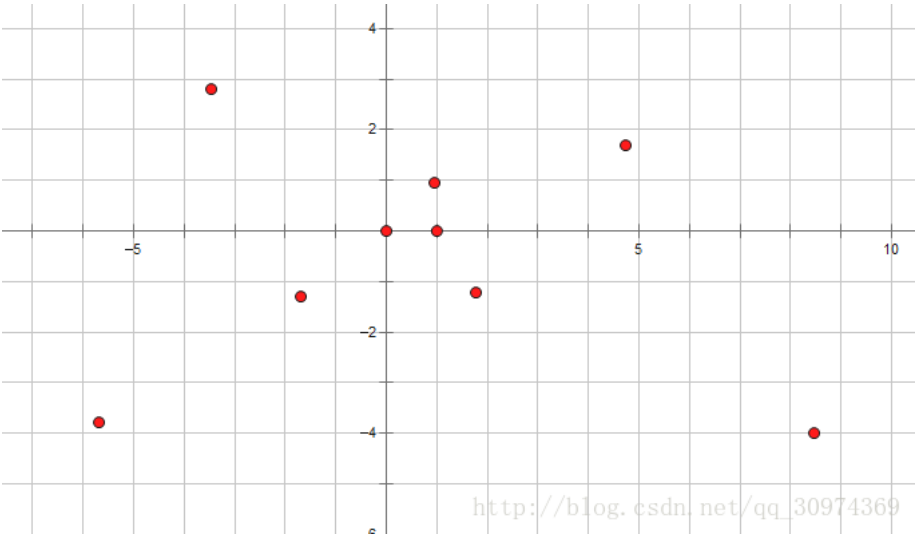
积分与排名

积分 - 286747

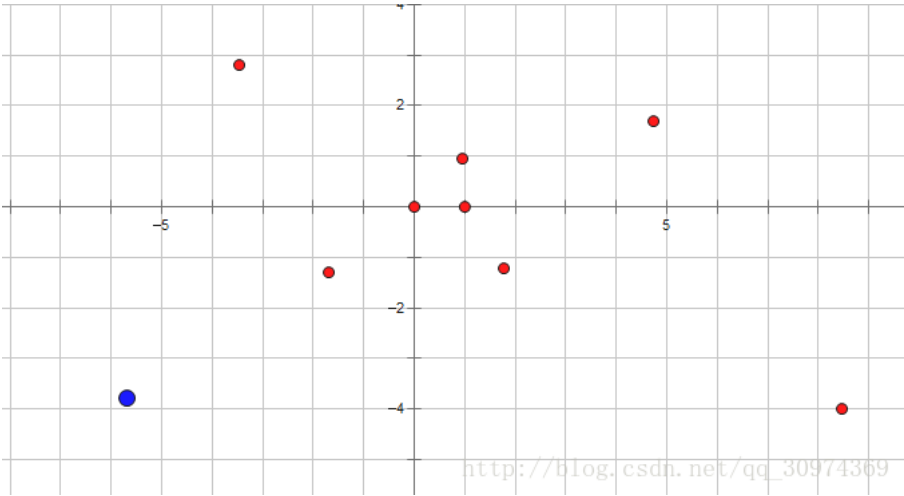
排名 - 1143

随笔分类

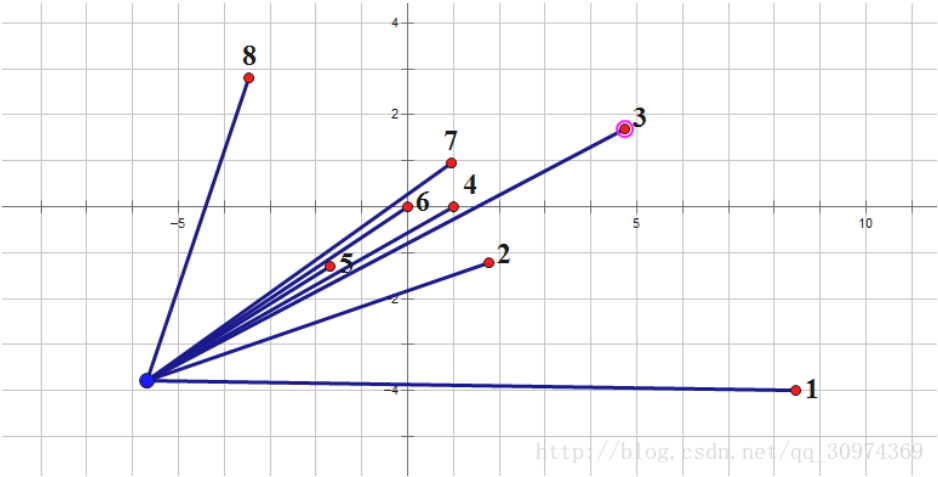
- A -- 模板(27)
- A -- 题解(70)
- A -- 游记&杂项(23)
- A -- 知识点(55)
- OJ -- 51NOD(4)
- OJ -- AtCoder(34)
- OJ -- BZOJ(667)
- OJ -- CJOJ(62)
- OJ -- CodeForces(88)
- OJ -- HDU(26)
- OJ -- Loj(40)
- OJ -- POJ(22)
- OJ -- TopCoder(1)
- OJ -- Uoj(48)
- OJ -- Vjudge(35)
- OJ -- 洛谷(167)
- OJ -- 牛客网(3)
- Source -- NOI(60)
- Source -- NOIIP(30)
- Source -- 各省省选(445)
- Source -- 网络流24题(23)
- 动态规划 -- 决策单调性(4)
- 动态规划 -- 轮廓线&插头(4)
- 动态规划 -- 凸优化(6)
- 动态规划 -- 斜率优化(11)
- 动态规划 -- 状态压缩(22)
- 多项式 -- FFT(38)
- 多项式 -- FWT(10)
- 多项式 -- 常系数齐次线性递推(3)
- 多项式 -- 多项式运算(12)
- 多项式 -- 拉格朗日插值(7)
- 多项式 -- 生成函数(13)
- 基础算法 -- 动态规划(287)
- 基础算法 -- 模拟(20)
- 基础算法 -- 三分(2)
- 基础算法 -- 搜索(35)
- 基础算法 -- 贪心(78)
- 计算几何 -- 半平面交(2)
- 计算几何 -- 闵可夫斯基和(1)
- 计算几何 -- 其他(4)
- 计算几何 -- 凸包(5)
- 计算几何 -- 辛普森积分(1)
- 计算几何 -- 旋转卡壳(2)
- 其他 -- CDQ分治(12)
- 其他 -- dsu on tree(2)
- 其他 -- 倍增(7)
- 其他 -- 边分治(3)
- 其他 -- 博弈论(26)



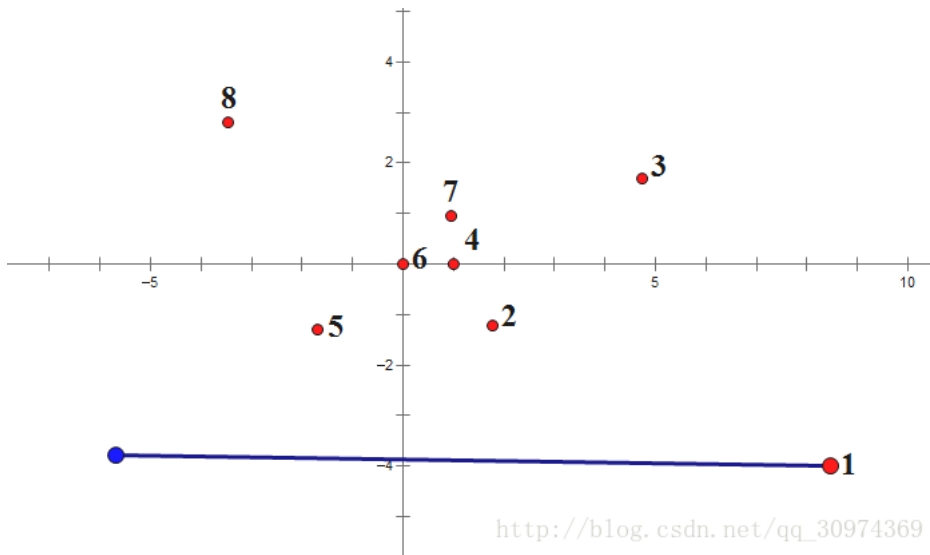
这是一片点。



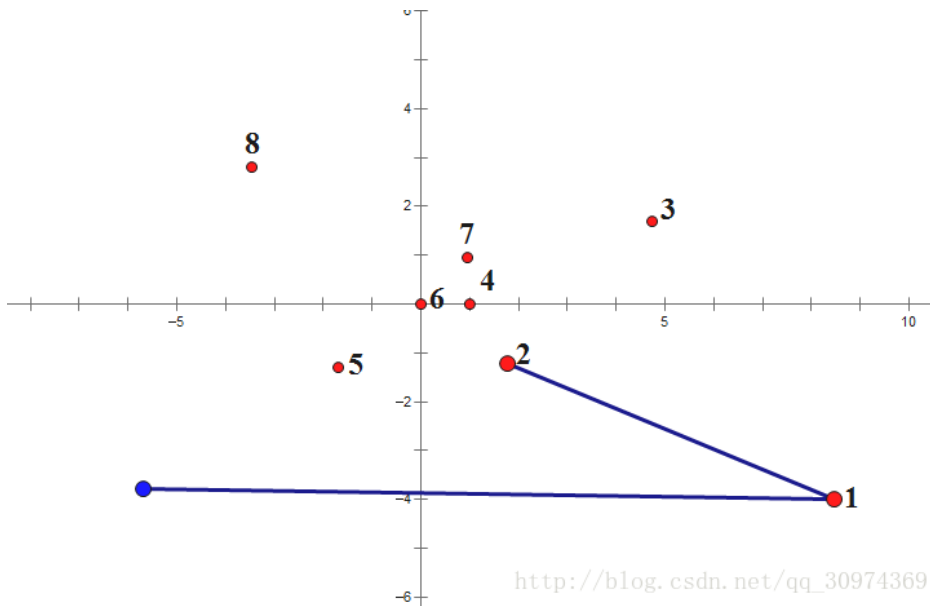
找到最靠近左下的一个点



其他的点按照极角排序

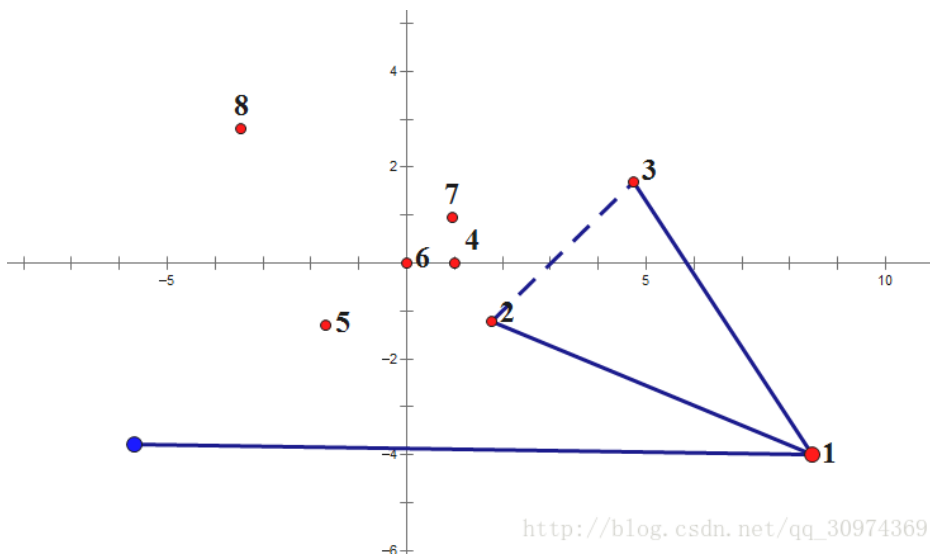


然后把1丢到凸包的栈里面，准备开始扫描



检查2号点是否在1的一侧，（检查一下是不是凸多边形）

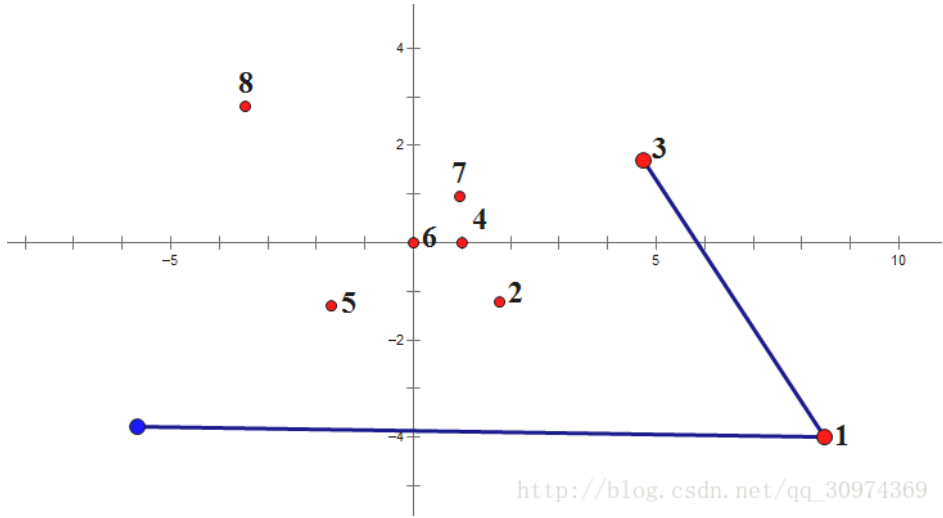
这里检查到2号可行，先加入到栈中



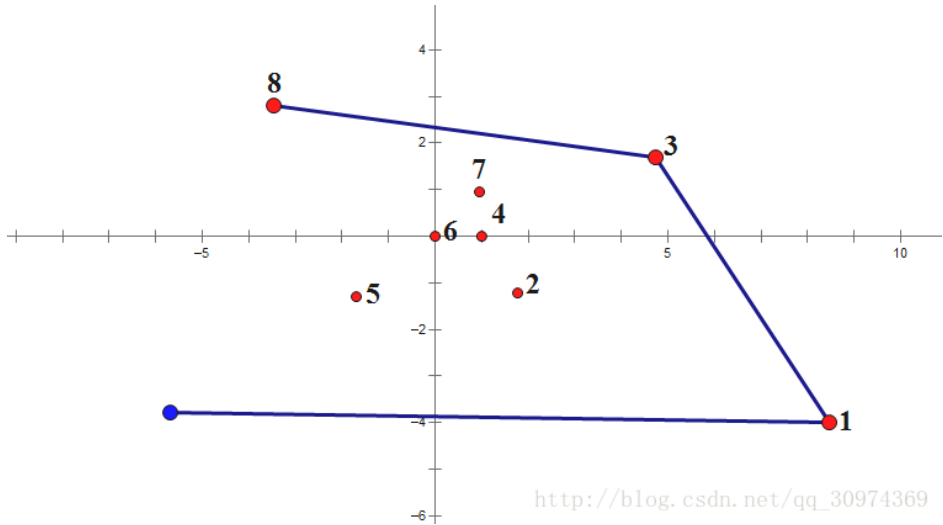
检查到3更加靠近外侧（如果加入3号就会形成凹多边形，显然3在凸包中，而2不在）

然后把2号点弹出栈，判断1号和3号节点的关系（同判断2号）

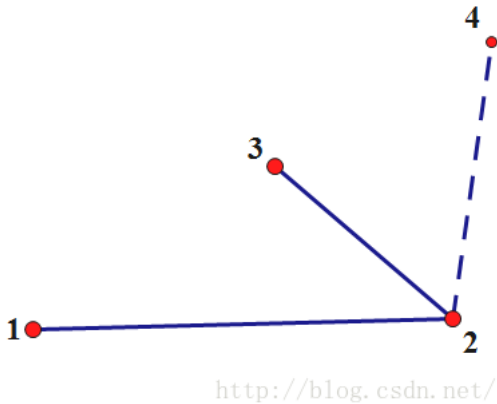
其他 -- 点分治(13)
 其他 -- 动态dp(3)
 其他 -- 二进制分治(4)
 其他 -- 分块(6)
 其他 -- 分数规划(13)
 其他 -- 模拟退火(3)
 其他 -- 莫队(9)
 其他 -- 启发式合并(3)
 其他 -- 扫描线(7)
 其他 -- 神仙题(4)
 其他 -- 树上莫队(3)
 其他 -- 线段树分治(7)
 其他 -- 虚树(8)
 其他 -- 折半搜索(4)
 其他 -- 整体二分(5)
 数据结构 -- KD-Tree(5)
 数据结构 -- Link-Cut Tree(24)
 数据结构 -- Splay(19)
 数据结构 -- ST表(3)
 数据结构 -- 并查集(17)
 数据结构 -- 长链剖分(7)
 数据结构 -- 单调队列(8)
 数据结构 -- 单调栈(7)
 数据结构 -- 动态点分治(7)
 数据结构 -- 哈夫曼树(1)
 数据结构 -- 李超线段树(2)
 数据结构 -- 树链剖分(19)
 数据结构 -- 树套树(10)
 数据结构 -- 树状数组(21)
 数据结构 -- 线段树(74)
 数据结构 -- 线性基(15)
 数据结构 -- 重构树(5)
 数据结构 -- 主席树(31)
 数据结构 -- 左偏树(5)
 数学方法 -- BSGS(5)
 数学方法 -- Burnside引理(4)
 数学方法 -- CRT(4)
 数学方法 -- Lucas定理(7)
 数学方法 -- min_25筛(5)
 数学方法 -- Pollard_rho(5)
 数学方法 -- prufer序列(1)
 数学方法 -- 单位根反演(3)
 数学方法 -- 杜教筛(9)
 数学方法 -- 二次剩余(1)
 数学方法 -- 高斯消元(9)
 数学方法 -- 矩阵快速幂(25)
 数学方法 -- 矩阵树定理(9)
 数学方法 -- 莫比乌斯反演(29)
 数学方法 -- 期望&概率(3)
 数学方法 -- 容斥原理(34)
 数学方法 -- 数论(56)
 数学方法 -- 斯特林数(13)
 数学方法 -- 斜率优化(13)
 数学方法 -- 组合计数(35)
 图论 -- 2-sat问题(7)
 图论 -- KM算法(2)
 图论 -- Tarjan(19)
 图论 -- 差分约束(6)
 图论 -- 带花树(3)
 图论 -- 二分图匹配(18)
 图论 -- 欧拉回路(5)
 图论 -- 平面图&对偶图(2)
 图论 -- 拓扑排序(4)
 图论 -- 网络流(85)
 图论 -- 仙人掌(5)
 图论 -- 弦图(2)
 图论 -- 匈牙利算法(11)
 图论 -- 圆方树(4)
 图论 -- 最短路(39)
 图论 -- 最小生成树(17)
 字符串算法 -- AC自动机(13)
 字符串算法 -- KMP算法(6)
 字符串算法 -- Manacher(5)
 字符串算法 -- Trie树(5)



依次这么判断，最后所有凸包上的点都会在栈中



这样子算法的步骤很显然了。
继续解决一些细节上的问题（貌似就一个把。。。）
怎么计算一个节点是否在前一个点的一侧。。。
~~（我说的好不专业。。。我自己都不知道该怎么说一些名词。。。就将就着理解一下吧。。。）~~
我们先拿几个点出来



其中1,2,3是当前在凸包的栈中的点，4号节点是需要判断的点
那么，我们需要从栈中拿最上方的两个点（2和3节点）
把他们连接起来，再把2和4连接起来（怎么连接？我是不会说直接用向量的坐标表示就可以了）
计算一下两个向量的叉积。。
哈，叉积。。
解释一下吧。。
假设2到3的向量是 $\mathbf{a}(x_1, y_1)$

字符串算法 -- 哈希(15)
字符串算法 -- 后缀数组(16)
字符串算法 -- 后缀自动机(29)
字符串算法 -- 回文树(10)

随笔档案

- 2019年7月(31)
- 2019年6月(28)
- 2019年5月(20)
- 2019年4月(59)
- 2019年3月(73)
- 2019年2月(45)
- 2019年1月(39)
- 2018年12月(53)
- 2018年11月(19)
- 2018年10月(92)
- 2018年9月(66)
- 2018年8月(75)
- 2018年7月(67)
- 2018年6月(33)
- 2018年5月(44)
- 2018年4月(79)
- 2018年3月(40)
- 2018年2月(70)
- 2018年1月(100)
- 2017年12月(44)
- 2017年11月(27)
- 2017年10月(54)
- 2017年9月(9)
- 2017年8月(31)
- 2017年7月(82)

最新评论

- 1. Re: AtCoder Grand Contest 006
Orzyyb
--菜狗xzz
- 2. Re: 【AtCoder3611】Tree MST（点分治，最小生成树）
Orzyyb
--Adscn
- 3. Re: NOI2019退役记 upd:2019.9.8
催更QwQ
--Adscn
- 4. Re: FWT快速沃尔什变换学习笔记
@ 小蒟蒻yyb就是“再来仔细的看着，这个东西也满足结合律。”这句话，这里应该是分配律而不是结合律吧。...
--Flower&)
- 5. Re: NOI2019退役记 upd:2019.9.8
歪歪比根脖
--smy☆

阅读排行榜

- 1. AC自动机(12226)
- 2. FWT快速沃尔什变换学习笔记(9468)
- 3. Splay入门解析【保证让你看不懂（滑稽）】(7847)
- 4. NOI2019退役记 upd:2019.9.8(5367)
- 5. NOI2018游记(5310)

评论排行榜

- 1. NOI2019退役记 upd:2019.9.8(82)
- 2. Splay入门解析【保证让你看不懂（滑稽）】(79)
- 3. NOI2018前的一些计划&记录(日更)(57)
- 4. yyb要填的各种总结的坑(55)
- 5. FWT快速沃尔什变换学习笔记(43)

推荐排行榜

- 1. WC2019冬眠记(50)

2到4的向量是 $\mathbf{b}(x_2, y_2)$

那么，计算一下它们的叉积，也就是 $x_1y_2 - x_2y_1$

换种方法表示就是。

$|\mathbf{a}| \cdot |\mathbf{b}| \cdot \sin \langle \mathbf{a}, \mathbf{b} \rangle$

（所以说叉积也可以用来求出三角形的面积~这个以后还会用到的）

如果，这两个向量的叉积 ≥ 0 证明这两个向量平行或者夹角是个锐角

也就证明了3号节点此时一定不再凸包上（因为连接2和4之后3在凸包内侧了）

把3号节点弹出栈，继续重复上面的步骤即可。

感觉我说的有点小复杂诶。。。

这个东东多画点图就会理解的

如果还是不太清楚，可以看一看代码。

```
struct Node
{
    int x, y;
} p[MAX], S[MAX]; // p储存节点的位置, s是凸包的栈
inline bool cmp(Node a, Node b) // 比较函数, 对点的极角进行排序
{
    double A = atan2((a.y - p[1].y), (a.x - p[1].x));
    double B = atan2((b.y - p[1].y), (b.x - p[1].x));
    if (A != B) return A < B;
    else return a.x < b.x; // 这里注意一下, 如果极角相同, 优先放x坐标更小的点
}
long long Cross(Node a, Node b, Node c) // 计算叉积
{
    return 1LL * (b.x - a.x) * (c.y - a.y) - 1LL * (b.y - a.y) * (c.x - a.x);
}
void Get() // 求出凸包
{
    p[0] = (Node) {INF, INF}; int k;
    for (int i = 1; i <= n; ++i) // 找到最靠近左下的点
        if (p[0].y > p[i].y || (p[0].y == p[i].y && p[i].x < p[0].x))
            {p[0] = p[i]; k = i;}
    swap(p[k], p[1]);
    sort(&p[2], &p[n+1], cmp); // 对于剩余点按照极角进行排序
    S[0] = p[1], S[1] = p[2]; top = 1; // 提前在栈中放入节点
    for (int i = 3; i <= n; ) // 枚举其他节点
    {
        if (top && Cross(S[top-1], p[i], S[top]) >= 0)
            top--; // 如果当前栈顶不是凸包上的节点则弹出
        else S[++top] = p[i++]; // 加入凸包的栈中
    }
    // 底下这个玩意用来输出凸包上点的坐标
    // for (int i = 0; i <= top; ++i)
    //     printf("(%d, %d)\n", S[i].x, S[i].y);
}
```

接下来找一道简单点的例题

[HDU 1392](#)

这道题目就是求出凸包然后计算周长，很简单的题目，去试试吧。。

分类: [A -- 模板](#) , [A -- 知识点](#)

好文要顶

关注我

收藏该文

🔖

🗨



小蒟蒻yyb
关注 - 36
粉丝 - 323

+加关注

« 上一篇: [POJ 3167 Layout \(差分约束\)](#)

» 下一篇: [凸包的直径——旋转卡壳](#)

3

0

posted @ 2017-07-30 21:22 小蒟蒻yyb 阅读(3132) 评论(0) 编辑 收藏