**凸包**

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);

}