

## 二维凸包模板 - kuangbin - 博客园

```
#include<stdio.h>
#include<math.h>
#include<algorithm>
#include<iostream>
using namespace std;
const int MAXN=1000;

struct point
{
    int x,y;
};
point list[MAXN];
int stack[MAXN],top;

int cross(point p0,point p1,point p2) //计算叉积 p0p1 X p0p2
{
    return (p1.x-p0.x)*(p2.y-p0.y)-(p1.y-p0.y)*(p2.x-p0.x);
}
double dis(point p1,point p2) //计算 p1p2的 距离
{
    return sqrt((double)(p2.x-p1.x)*(p2.x-p1.x)+(p2.y-p1.y)*(p2.y-p1.y));
}
bool cmp(point p1,point p2) //极角排序函数， 角度相同则距离小的在前面
{
    int tmp=cross(list[0],p1,p2);
    if(tmp>0) return true;
    else if(tmp==0&&dis(list[0],p1)<dis(list[0],p2)) return true;
    else return false;
}
void init(int n) //输入，并把 最左下方的点放在 list[0]。并且进行极角排序
{
    int i,k;
    point p0;
    scanf("%d%d",&list[0].x,&list[0].y);
    p0.x=list[0].x;
    p0.y=list[0].y;
    k=0;
    for(i=1;i<n;i++)
    {
        scanf("%d%d",&list[i].x,&list[i].y);
        if( (p0.y>list[i].y) || ((p0.y==list[i].y)&&(p0.x>list[i].x)) )
        {
            p0.x=list[i].x;
            p0.y=list[i].y;
            k=i;
        }
    }
    list[k]=list[0];
    list[0]=p0;

    sort(list+1,list+n,cmp);
}

void graham(int n)
{

```

```
int i;
if(n==1) {top=0;stack[0]=0;}
if(n==2)
{
    top=1;
    stack[0]=0;
    stack[1]=1;
}
if(n>2)
{
    for(i=0;i<=1;i++) stack[i]=i;
    top=1;

    for(i=2;i<=n;i++)
    {
        while(top>0&&cross(list[stack[top-1]],list[stack[top]],list[i])<=0) top--;
        top++;
        stack[top]=i;
    }
}
}
```