

## 计算几何

### 极角排序

```
struct Point { double x,y; };
double cross(double x1,double y1,double x2,double y2)
{
    return (x1*y2-x2*y1);
}
double compare(const Point& a,const Point& b,const Point& c)
{
    return cross((b.x-a.x),(b.y-a.y),(c.x-a.x),(c.y-a.y));
}
bool cmp(const Point& p,const Point& q)
{
    Point O = {0,0};
    if(compare(O,p,q)==0) return p.x<q.x;
    else return compare(O,p,q)>0;
}
int quadrant(const Point& p)
{
    if(p.x>0&&p.y>=0) return 1;
    if(p.x<=0&&p.y>0) return 2;
    if(p.x<0&&p.y<=0) return 3;
    if(p.x>=0&&p.y<0) return 4;
}
bool operator<(const Point& p, const Point& q)
{
    if(quadrant(p)==quadrant(q)) return cmp(p,q);
    return quadrant(p)<quadrant(q);
}
// atan2 法
// bool operator<(const Point& p,const Point& q)
// {
//     if(atan2(p.y,p.x)!=atan2(q.y,q.x)) return atan2(p.y,p.x)<atan2(q.y,q.x);
//     else return p.x<q.x;
// }
```

### 欧拉公式

顶点数-棱长数 + 表面数 =2