旋转卡壳

Rotating Calipers

Idea：首先求出凸包，随后逆时针枚举边，利用三角形面积寻找最远点，容易发现最远点的轨迹也是逆时针的，该最远点可能是边的两个端点的对踵点对，由此可求出所有对踵点对。

Application：求凸包直径、宽度，凸包间最大、小距离，最小面积、周长外接矩形，洋葱、螺旋三角剖分，四边形剖分，合并凸包、凸包公切线、凸包交集、凸包临界切线、凸多边形矢量和，最薄横截带

Reference：链接

Complexity：$O(n)$（仅就旋转卡壳而言；事实上，由于一般需要先求凸包，复杂度是凸包的复杂度）

Code（求凸包直径的平方）：

```
int ans;
void RotatingCalipers(int m, Point p[]){ // p[] = sta[], m = staid in ConvexHull()
    if(m == 2){
        ans = (int)( (p[1] - p[2]) * (p[1] - p[2]) );
        return;
    }
    p[m+1] = p[1];
    int ver = 2;
    for(int i = 1; i <= m; i++){
        while(TriangleArea(p[i], p[i+1], p[ver]) < TriangleArea(p[i], p[i+1], p[ver+1])){
            ver++;
            if(ver == m+1)    ver = 1;
            ans = max(ans, (int)max((p[ver] - p[i]) * (p[ver] - p[i]), (p[ver] - p[i+1]) * (p[ver] - p[i+1])));
        }
    }
}
```