最小圆覆盖

Minimum Enclosing Circle

Idea：随机增量法。假设圆 O 是前 $i-1$ 个点的最小覆盖圆，若第 i 个点在圆内或圆上，不用处理；否则，新圆应该经过第 i 个点。我们现在以第 1 个点和第 i 个点为直径作圆，但是这个圆可能包含不了所有前 $i-1$ 个点，设第 j 个点在圆外，那么新圆一定经过第 j 个点，以第 i 个点和第 j 个点为直径作圆，同理可得到新圆一定经过第 k 个点。三点确定一个圆，我们就找到了覆盖前 i 个点的最小圆。随机化以避免最坏情形出现。

Complexity：期望 $O(n)$

Code：

```
Point getO(Point a, Point b, Point c){
    Point mab = (a + b) / 2, mbc = (b + c) / 2;
    Vector vab = Normal(b - a), vbc = Normal(c - b);
    return GetLineIntersection(Line(mab, vab), Line(mbc, vbc));
}
Circle minEnclosingCircle(int n, Point p[]){
    random_shuffle(p+1, p+n+1);
    Point o = p[1]; double r = 0;
    for(int i = 1; i <= n; i++){
        if(cmp(Length(o - p[i]), r) <= 0)   continue;
        o = (p[i] + p[1]) / 2;
        r = Length(p[i] - p[1]) / 2;
        for(int j = 1; j < i; j++){
            if(cmp(Length(o - p[j]), r) <= 0)   continue;
            o = (p[j] + p[i]) / 2;
            r = Length(p[j] - p[i]) / 2;
            for(int k = 1; k < j; k++){
                if(cmp(Length(o - p[k]), r) <= 0)   continue;
                o = getO(p[i], p[j], p[k]);
                r = Length(p[i] - o);
            }
        }
    }
    return Circle(o, r);
}
```