

```

1 #include "geometry.hpp"
2 #include "line.hpp"
3
4 using Polygon = std::vector<Point>;
5
6 ld area(const Polygon &p)
7 {
8     ld res = 0;
9     int n = p.size();
10    for (int i = 0; i < n; i++)
11    {
12        res += cross(p[i], p[(i + 1) % n]);
13    }
14    return res / 2;
15 }
16
17 bool is_counter_clockwise(const Polygon &poly)
18 {
19     ld angle = 0;
20     int n = poly.size();
21     for (int i = 0; i < n; i++)
22     {
23         Point a = poly[i], b = poly[(i + 1) % n], c = poly[(i + 2) % n];
24         angle += arg((c - b) / (b - a));
25     }
26     return angle > eps;
27 }
28
29 // -1 => out
30 // 0 => on
31 // 1 => in
32 int is_in_polygon(const Polygon &poly, Point p)
33 {
34     ld angle = 0;
35     int n = poly.size();
36     for (int i = 0; i < n; i++)
37     {
38         Point a = poly[i], b = poly[(i + 1) % n];
39         if (isis_sp(Line(a, b), p))
40             return 1;
41         angle += arg((b - p) / (a - p));
42     }
43     return eq(angle, 0) ? 0 : 2;
44 }
45
46 // 凸包
47 Polygon convex_hull(std::vector<Point> ps)
48 {
49     int n = ps.size();
50     int k = 0;
51     sort(ps.begin(), ps.end());
52     Polygon ch(2 * n);
53     for (int i = 0; i < n; ch[k++] = ps[i++])
54         while (k >= 2 && ccw(ch[k - 2], ch[k - 1], ps[i]) <= 0)
55             --k;
56     for (int i = n - 2, t = k + 1; i >= 0; ch[k++] = ps[i--])
57         while (k >= t && ccw(ch[k - 2], ch[k - 1], ps[i]) <= 0)
58             --k;
59     ch.resize(k - 1);
60     return ch;
61 }
62
63 // 凸包カット
64 Polygon convex_cut(const Polygon &ps, Line l)
65 {
66     int n = ps.size();
67     Polygon Q;
68     for (int i = 0; i < n; i++)
69     {
70         Point A = ps[i], B = ps[(i + 1) % n];
71         Line m = Line(A, B);
72         if (ccw(l.a, l.b, A) != -1)
73             Q.emplace_back(A);
74         if (ccw(l.a, l.b, A) * ccw(l.a, l.b, B) < 0 && isis_ll(l, m))
75             Q.emplace_back(isis_ll(l, m));
76     }
77     return Q;
78 }
79
80 void add_point(std::vector<Point> &ps, Point p)
81 {
82     for (Point q : ps)
83         if (abs(q - p) < eps)
84             return;
85     ps.emplace_back(p);
86 }
87
88 // Voronoi 図: http://judge.u-aizu.ac.jp/onlinejudge/review.jsp?rid=3668810
89 // polygon を ps でボロノイ分割する時の ps[index] に割り当てられる多角形
90 Polygon voronoi(const Polygon &polygon, const vector<Point> &ps, int index)
91 {
92     Polygon p = polygon;
93     int n = ps.size();
94     for (int j = 0; j < n; j++)
95     {

```

```
96         if (index == j)
97             continue;
98         Line l = bisector(ps[index], ps[j]);
99         p = convex_cut(p, l);
100     }
101     return p;
102 }
103
```