

## C5F 寄蒜几盒V

---

### 题目描述

求凸包模板题

### 解题思路

- [Andrew 算法](#)
- [Graham 算法](#)

排序时按横坐标第一关键字，纵坐标第二关键字

求凸包时升序枚举求出下凸壳，降序求出上凸壳

求凸壳时，一旦发现即将进栈的点 ( $P$ ) 和栈顶的两个点  $S_1, S_2$  行进的方向向右旋转，即叉积小于0:  $\vec{S_2S_1} \times \vec{S_1P} < 0$  则弹出栈顶，回到上一步，继续检测，直到  $\vec{S_2S_1} \times \vec{S_1P} \geq 0$  或者栈内仅剩一个元素为止。

### 代码实现

```
#include <bits/stdc++.h>

using namespace std;

using ll = long long;
using db = long double; // or double, if TL is tight
using str = string;
using u32 = unsigned int;
using u64 = unsigned long long;
using u128 = unsigned __int128;

using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<db, db>;
#define fi first
#define se second

#define vc vector
using vi = vc<int>;
using vb = vc<bool>;
using vll = vc<ll>;
using vd = vc<db>;
using vs = vc<str>;
using vch = vc<char>;
using vpii = vc<pii>;
```

```

using vp1l = vc<p1l>;
using vpdd = vc<pdd>;
using vvi = vc<vc<int>>;
using vv1l = vc<vc<1l>>;
using vvch = vc<vc<char>>;
using vvb = vc<vc<bool>>;
using vvp1i = vc<vc<p1i>>;
using vvp1l = vc<vc<p1l>>;
using vvvi = vc<vc<vc<int>>>;

const int mod = 998244353; // 1e9 + 7;
const int INF = 0x3f3f3f3f;
const 1l BIG = 1e18; // not too close to LLONG_MAX
const db PI = acos((db)-1);
const int dx[4]{1, 0, -1, 0}, dy[4]{0, 1, 0, -1}; // for
every grid problem!!
mt19937
rng32((uint32_t)chrono::steady_clock::now().time_since_epo
ch().count());
mt19937_64
rng64((uint64_t)chrono::steady_clock::now().time_since_epo
ch().count());

#define cl(x, y) memset(x, y, sizeof(x))
#define de(x) cerr << #x << " = " << x << " "
#define del(x) cerr << #x << " = " << x << endl

#define overload3(a, b, c, d, ...) d
#define rep1(a) for (int i = 0; i < (a); i++)
#define rep2(i, a) for (int i = 0; i < (a); i++)
#define rep3(i, a, b) for (int i = (a); i <= (b); i++)
#define per1(a) for (int i = (a) - 1; i >= 0; i--)
#define per2(i, a) for (int i = (a) - 1; i >= 0; i--)
#define per3(i, a, b) for (int i = (a); i >= (b); i--)
#define rep(args...) overload3(args, rep3, rep2, rep1)
(args)
#define per(args...) overload3(args, per3, per2, per1)
(args)
#define for_subset(t, s) \
    for (1l t = (s); t >= 0; t = (t == 0 ? -1 : (t - 1) &
(s)))

#define co cout
#define dl '\n'
#define ret return
#define pb push_back
#define tcT template <class T
#define YES cout << "YES" << endl
#define Yes cout << "Yes" << endl
#define NO cout << "NO" << endl
#define No cout << "No" << endl
#define INT(args...) \

```

```

    int args;          \
    rd(args)
#define LL(args...) \
    ll args;          \
    rd(args)

#define all(s) s.begin(), s.end()
#define rall(s) s.rbegin(), s.rend()
#define sz(v) int(v.size())
#define SUM(a) accumulate(all(a), 0ll)
#define MIN(v) (*min_element(all(v)))
#define MAX(v) (*max_element(all(v)))
#define unisort(v) \
    sort(all(v)); \
    v.resize(unique(all(v)) - v.begin())

int popcnt(int x) { return __builtin_popcount(x); }
int popcnt(u32 x) { return __builtin_popcount(x); }
int popcnt(ll x) { return __builtin_popcountll(x); }
int popcnt(u64 x) { return __builtin_popcountll(x); }

#define LOWBIT(x) ((x) & (-x))
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 1, 2)
int topbit(int x) { return (x == 0 ? -1 : 31 -
__builtin_clz(x)); }
int topbit(u32 x) { return (x == 0 ? -1 : 31 -
__builtin_clz(x)); }
int topbit(ll x) { return (x == 0 ? -1 : 63 -
__builtin_clzll(x)); }
int topbit(u64 x) { return (x == 0 ? -1 : 63 -
__builtin_clzll(x)); }
// (0, 1, 2, 3, 4) -> (-1, 0, 1, 0, 2)
int lowbit(int x) { return (x == 0 ? -1 :
__builtin_ctz(x)); }
int lowbit(u32 x) { return (x == 0 ? -1 :
__builtin_ctz(x)); }
int lowbit(ll x) { return (x == 0 ? -1 :
__builtin_ctzll(x)); }
int lowbit(u64 x) { return (x == 0 ? -1 :
__builtin_ctzll(x)); }
tcT > T cdiv(T &a, T &b)
{
    return a / b + ((a ^ b) > (T)(0) && a % b);
} // divide a by b rounded up
tcT > T fdiv(T &a, T &b)
{
    return a / b - ((a ^ b) < (T)(0) && a % b);
} // divide a by b rounded down

tcT, class S > inline ostream &operator<<(ostream &i,
const pair<T, S> &a) { return i << a.first << ' ' <<
a.second; }

```

```

tcT, class S > inline istream &operator>>(istream &i,
pair<T, S> &a) { return i >> a.first >> a.second; }
tcT > inline ostream &operator<<(ostream &i, const
vector<T> &a)
{
    if (a.size())
        i << a[0];
    for (int j = 1; j < int(a.size()); j++)
        i << ' ' << a[j];
    return i;
}
tcT > inline istream &operator>>(istream &i, vector<T> &a)
{
    for (auto &j : a)
        i >> j;
    return i;
}
void err(istream_iterator<string> it) {
it.~istream_iterator(); }
template <typename T, typename... Args>
void err(istream_iterator<string> it, T a, Args... args)
{
    cerr << *it << " = " << a << endl;
    err(++it, args...);
}
void wt() { cout << '\n'; }
tcT, class... Args > void wt(const T &a, const Args
&...args)
{
    cout << a;
    (void)(cout << ... << (cout << ' ', args));
    cout << '\n';
}
tcT > void wt(vector<vector<T>> &a)
{
    for (auto &i : a)
        wt(i);
}
tcT, class S > void wt(vector<pair<T, S>> &a)
{
    for (auto &i : a)
        wt(i);
}
template <class... T>
void rd(T &...a) { (cin >> ... >> a); }

#define LOCAL_TEST freopen("out.txt", "w", stdout)
#define IOS \
    ios::sync_with_stdio(false); \
    cin.tie(0); \
    cout.tie(0)
#define FIX(x) cout << fixed << setprecision(x)

```

```

const int maxn = 1e5 + 5;

inline void INIT()
{
}

/*

*/

template <class T>
struct Point {
    typedef Point P;
    T x, y;

    explicit Point(T x = 0, T y = 0) : x(x), y(y) {}
    bool operator<(P p) const { return tie(x, y) <
tie(p.x, p.y); }
    bool operator==(P p) const { return tie(x, y) ==
tie(p.x, p.y); }
    P operator+(P p) const { return P(x + p.x, y + p.y); }
    P operator-(P p) const { return P(x - p.x, y - p.y); }
    P operator*(T d) const { return P(x * d, y * d); }
    P operator/(T d) const { return P(x / d, y / d); }
    T dot(P p) const { return x * p.x + y * p.y; }
    T cross(P p) const { return x * p.y - y * p.x; }
    T cross(P a, P b) const { return (a - *this).cross(b -
*this); }
    T dist2() const { return x * x + y * y; }
    double dist() const { return sqrt((double)dist2()); }
};

/*
    Returns a vector of the points of the convex hull in
counterclockwise order.
    Points on the edge of the hull between two
other points are not considered part of the hull.
*/
template<class T>
vector<Point<T>> convexHull(vector<Point<T>> pts) {
    if (sz(pts) <= 1) return pts;
    sort(all(pts));
    vector<Point<T>> h(sz(pts) + 1);
    int s = 0, t = 0;
    for (int it = 2; it--; s = --t, reverse(all(pts)))
        for (Point<T> p : pts) {
            while (t >= s + 2 && h[t - 2].cross(h[t - 1], p)
<= 0) t--;
            h[t++] = p;
        }
}

```

```

        return {h.begin(), h.begin() + t - (t == 2 && h[0] ==
h[1])};
    }

    inline void SOLVE() {

        INT(n);
        vc<Point<long double>> s(n);

        rep(n) {
            cin >> s[i].x >> s[i].y;
        }

        vc<Point<long double>> res = convexHull(s);
        Point<long double> zero(0, 0);
        long double ans = 0;
        if(sz(res) >= 1) { ans += zero.cross(res[sz(res) - 1],
res[0]); }
        rep(i, sz(res) - 1)
            ans += zero.cross(res[i], res[i + 1]);
        ans = fabs(ans);
        wt(ans / 2);

    }

    int main()
    {

        IOS; FIX(1);

        INIT();
        int tt = 1;
        cin >> tt;
        while (tt--)
            SOLVE();

        // cout << "program ends..." << endl;
        // system("pause");
        return 0;
    }

```