

E5I 寄蒜几盒IX

题目描述

给定 n 个点，每个点拥有坐标 (x_i, y_i) ，对于每个 i 求：

$$E_i = \frac{1}{n-1} \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{x_i x_j \cdot y_i y_j}{i-j}$$

解题分析

对于原式，我们进行小小的变换

$$\begin{aligned} E_i &= \frac{1}{n-1} \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{x_i x_j \cdot y_i y_j}{i-j} \\ &= \frac{x_i y_i}{n-1} \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{x_j \cdot y_j}{i-j} \end{aligned}$$

即，我们对于每个 i 进行简化，仅需求 \sum 中的值，最后进行一个附加即可

对于 \sum 中的部分，通过观察可以看到分子 $x_j y_j$ 的下标 + 分母 = i 为定值

故，构造多项式

$$\begin{aligned} A_i X &= x_{n-i-1} y_{n-i-1} X, 0 \leq i \leq n-1 \\ B_i X &= \frac{1}{n-i-1} X, 0 \leq i \leq 2n-2 \end{aligned}$$

故多项式乘积的第 i 项与 E 的第 $2n-2-i$ 项对应

故我们仅需对于上式进行快速傅里叶变换，取结果系数的 $n-1 \sim 2n-2$ 项的值即为我们要求的简化后的 E 数组

时间复杂度： $O(n \log n)$

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 vpri = vc<pii>;
using vpll = vc<pll>;
using vpdd = vc<pdd>;
using vvi = vc<vc<int>>;
using vvll = vc<vc<ll>>;
using vvch = vc<vc<char>>;
using vvbool = vc<vc<bool>>;
using vvpii = vc<vc<pii>>;
using vvpll = vc<vc<pll>>;
using vvvi = vc<vc<vc<int>>>;

const int mod = 998244353; // 1e9 + 7;
const int INF = 0x3f3f3f3f;
const ll 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 (ll 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() {

}

struct Complex {
    db x, y;

    Complex(db _x = 0.0, db _y = 0.0) { x = _x, y = _y; }

    Complex operator-(const Complex &b) const { return
Complex(x - b.x, y - b.y); }
    Complex operator+(const Complex &b) const { return
Complex(x + b.x, y + b.y); }
    Complex operator*(const Complex &b) const { return
Complex(x * b.x - y * b.y, x * b.y + y * b.x); }
};

/**
 * inversion transform before FFT and IFFT
 * length MUST be power of 2
 * @param y polynomial
 */
typedef vc<Complex> C;
void change(C &y) {
    int n = sz(y);
    for(int i = 1, j = n / 2; i < n - 1; i++) {
        if(i < j) swap(y[i], y[j]);
        int k = n / 2;
        while(j >= k) { j -= k; k >>= 1; }
        if(j < k) j += k;
    }
}

/**
 * FFT
 * n MUST be power of 2
 * when on == 1, DFT. on == -1, IDFT
 * @param y polynomial
 * @param on DFT/IDFT
 */
void FFT(C &y, int on = 1) {
    int n = sz(y);

```

```

change(y);
for(int h = 2; h <= n; h <= 1) {
    Complex wn(cos(2 * PI / h), sin(2 * PI / h));
    for(int j = 0; j < n; j += h) {
        Complex w(1, 0);
        for(int k = j; k < j + h / 2; k++, w = w * wn)
        {
            Complex u = y[k], t = w * y[k + h / 2];
            y[k] = u + t;
            y[k + h / 2] = u - t;
        }
    }
}
if(on == -1)
    rep(n) y[i].x /= n;
}

/**
 * extend the length to power of 2
 * store length in &lim
 */
void extend(C &a, C &b, int &lim) {
    lim = 1; int n = sz(a), m = sz(b);
    while(lim <= n + m) { lim <= 1; }
    a.resize(lim, Complex());
    b.resize(lim, Complex());
}

inline void SOLVE() {

    INT(n);
    vpii a(n); cin >> a;
    vc<Complex> A(2 * n - 1), B(n);

    rep(i, 2 * n - 1)
        if(n - i - 1 != 0)
            A[i].x = 1.0 / (n - i - 1);
    rep(i, n)
        B[i].x = a[n - i - 1].first * a[n - i - 1].second;

    int lim;
    extend(A, B, lim);

    FFT(A, 1); FFT(B, 1);
    rep(i, lim) A[i] = A[i] * B[i];
    FFT(A, -1);

    // cout << -0.0005 << " " << -0.00049 << " " << 0.0 <<
endl;

    // 0.0005
    rep(n) {

```

```

        db ans = (a[i].first * a[i].second * 1.0) / (n -
1) * A[2 * n - 2 - i].x;
        if(abs(ans) < 0.0005) cout << 0.0 << " ";
        else cout << ans << " ";
    }
    cout << endl;
}

/*

*/

int main() {

    IOS; FIX(3);

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

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

```