Beijing Normal University

Template

Pray Never

2024 年 11 月 20 日

目录

1 geometry

1.1 convex.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  struct Point {
     ll x,y;
  };
  auto andrew = [](vector<Point>& p) -> vector<Point> {    // 传入下标从零开始的点数组，返回凸包数组
    auto cmp = [](Point &a, Point &b) -> bool {
      if(a.x != b.x) return a.x < b.x;
      return a.y < b.y;
    };
    auto cross = [](Point &u, Point &v, Point &w) -> bool {
      ll x1 = u.x - v.x, y1 = u.y - v.y;
      ll x2 = w.x - v.x, y2 = w.y - v.y;
      return x1 * y2 - x2 * y1 > 0; //如果不希望在凸包的边上有输入点。把 > 改成 >=
    };
    sort(p.begin(), p.end(), cmp);
    int n = p.size(), m = 0;
    vector<Point> res(n + 1);
    for(int i = 0; i < n; ++i){
      while(m > 1 && !cross(res[m - 1],res[m - 2], p[i])) --m;
      res[m++] = p[i];
    }
    int kk = m;
    for(int i = n - 2; i >= 0; i--){
      while(m > kk && !cross(res[m - 1], res[m - 2], p[i])) --m;
      res[m++] = p[i];
    }
    if(n > 1) --m;//凸包有 m 个顶点
    res.erase(res.begin() + m, res.end());
    return res;
  };
#endif
```

2 graph

2.1 LCA.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  vector<vector<int> > a(n + 1, vector<int>(20)), v(n + 1);
  vector<int> dep(n + 1);
  auto build = [&](int u, int fa, auto&& self) -> void {
    dep[u] = dep[fa] + 1, a[u][0] = fa;
    for(int i = 1; i <= 19; ++i)
      a[u][i] = a[a[u][i - 1]][i - 1];
    for(int i: v[u]){
      if(i == fa) continue;
      self(i, u, self);
    }
  };
  auto lca = [&](int x, int y) -> int {
    if(dep[y] > dep[x]) swap(x, y);
    for(int i = 19; i >= 0; --i){
      if(dep[a[x][i]] >= dep[y])
        x = a[x][i];
    }
    if(x == y) return x;
    for(int i = 19; i >= 0; --i){
      if(a[x][i] != a[y][i])
        x = a[x][i], y = a[y][i];
    }
    return a[x][0];
  };
#endif



```

2.2 tarjan.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  int dfn_cnt = 0, scc_cnt = 0;
  vector<int> dfn(n + 1), low(n + 1), scc_id(n + 1);
  stack<int> s;
  vector<bool> in_stack(n + 1);
  vector<vector<int> > v(n + 1), scc(n + 1);
  // scc_id 是每个节点所属于的 scc 编号, scc 是这个编号下的所有节点
  auto tarjan = [&](int u, auto&& self) -> void {
    dfn[u] = low[u] = ++dfn_cnt;
    s.push(u), in_stack[u] = true;
    for(int i: v[u]){
      if(!dfn[i]){
        self(i, self);
        low[u] = min(low[u], low[i]);
      }
      else if(in_stack[i])
        low[u] = min(low[u], dfn[i]);
    }
    if(dfn[u] == low[u]){
      int tp;
      ++scc_cnt;
      do{
        tp = s.top();
        scc_id[tp] = scc_cnt;
        scc[scc_cnt].push_back(tp);
        in_stack[tp] = false;
        s.pop();
      } while(tp != u);
    }
  };
#endif
```

3 heading

3.1 debug.h

```
#include <bits/stdc++.h>
#define typet typename T
#define typeu typename U
#define types typename... Ts
#define tempt template <typet>
#define tempu template <typeu>
#define temps template <types>
#define tandu template <typet, typeu>

tandu std::ostream& operator<<(std::ostream& os, const std::pair<T, U>& p) {
return os << '<' << p.ff << ',' << p.ss << '>';
}
template <
typet, typename = decltype(std::begin(std::declval<T>())),
typename = std::enable_if_t<!std::is_same_v<T, std::string>>>
std::ostream& operator<<(std::ostream& os, const T& c) {
auto it = std::begin(c);
if (it == std::end(c)) return os << "{}";
for (os << '{' << *it; ++it != std::end(c); os << ',' << *it);
return os << '}';
}
#define debug(arg...) \
do { \
 std::cerr << "[" #arg "] :"; \
 dbg(arg); \
}while(false)

temps void dbg(Ts... args) {
(..., (std::cerr << ' ' << args));
std::cerr << '\n';
}

```

3.2 duipai.cpp

```
#ifdef IGNORE_THIS_FILE
  system("g++ -std=c++2a wa.cpp -o/wa");
  system("g++ -std=c++2a ac.cpp -o/ac");
  system("g++ -std=c++2a gen.cpp -o/gen");
  for(int i = 1; i <= 50; i++){
    std::cerr << "Test" << i << " : ";
    system("./gen > gen.in");
    system("./ac < gen.in > ac.out");
    system("./wa < gen.in > wa.out");
    if (system("diff ac.out wa.out")) {
      std::cerr << "ERR\n";
```

```
        return 0;
      }
      std::cerr << "AC\n";
    }
#endif
```

3.3 heading.cpp

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using i128 = __int128;
#define ff first
#define ss second
#include "debug.h"
constexpr int mod = 998244353;
constexpr ll INF = 1e18;
constexpr double pi = 3.141592653589793;
constexpr double eps = 1e-6;


```

4 math

4.1 Eratost.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  auto sieve = [](int n) -> vector<int> {
    vector<bool> is_prime(n + 1);
    vector<int> prime;
    for(int i = 2; i <= n; ++i){
      is_prime[i] = true;
    }
    for(int i = 2; i * i <= n; ++i){
      if(is_prime[i]){
        for(int j = i * i; j <= n; j += i)
          is_prime[j] = false;
      }
    }
    for(int i = 2; i <= n; ++i){
      if(is_prime[i])
        prime.push_back(i);
    }
    return prime;
  };
#endif
```

4.2 Euler.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  // vector<int> fac(n + 1);
  auto sieve = [&](int n) -> vector<int> {
    vector<int> prime;
    vector<bool> no_prime(n + 1);
    for(int i = 2; i <= n; ++i){
      if(!no_prime[i]){
        prime.push_back(i);
        // fac[i] = i;
      }
      for(int j: prime){
        if(j * i > n) break;
        no_prime[j * i] = true;
        // fac[j * i] = j;
        if(i % j == 0) break;
      }
    }
    return prime;
  };
#endif
```

4.3 pollar.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  using ll = long long;
  using ull = unsigned long long;
  bool is_prime(ull n) {
    if(n == 2) { return true; }
    if(n % 2 == 0) { return false; }
    auto internal_pow = [&](ull x, ull y) {
      ull r = 1;
      __uint128_t c = x;
      for(; y; y >>= 1, c = c * c % n) {
        if(y & 1) { r = __uint128_t(r) * c % n; }
      }
      return r;
    };
    auto MillerRabin = [&](ull a) {
      if(n <= a) { return true; }
      int e = __builtin_ctzll(n - 1);
      ull z = internal_pow(a, (n - 1) >> e);
      if(z == 1 || z == n - 1) { return true; }
      while(--e) {
        z = __uint128_t(z) * z % n;
        if(z == 1) { return false; }
        if(z == n - 1) { return true; }
      }
      return false;
    };
    vector<ull> cur;
    if(n < 4759123141) cur = vector<ull>{2, 7, 61};
    else cur = vector<ull>{2, 325, 9375, 28178, 450775, 9780504, 1795265022};
    return all_of(cur.begin(), cur.end(), [&](auto x) { return MillerRabin(x); });
  }

  struct Montgomery {
    ull mod, R;
    public:
    Montgomery(ull n): mod(n), R(n) {
      for(int i = 0; i < 5; i++) { R *= 2 - mod * R; }
    }
    ull fma(ull a, ull b, ull c) const {
      const __uint128_t d = __uint128_t(a) * b;
      const ull e = c + mod + (d >> 64);
      const ull f = ull(d) * R;
      const ull g = (__uint128_t(f) * mod) >> 64;
      return e - g;
    }
    ull mul(ull a, ull b) const { return fma(a, b, 0); }
  };
  ull PollardRho(ull n) {
    if(n % 2 == 0) { return 2; }
    const Montgomery m(n);
    constexpr ull C1 = 1, C2 = 2, M = 512;
    ull Z1 = 1, Z2 = 2;
  retry:
    ull z1 = Z1, z2 = Z2;
    for(unsigned k = M;; k <<= 1) {
      const ull x1 = z1 + n, x2 = z2 + n;
      for(unsigned j = 0; j < k; j += M) {
        const ull y1 = z1, y2 = z2;
        ull q1 = 1, q2 = 2;
        z1 = m.fma(z1, z1, C1), z2 = m.fma(z2, z2, C2);
        for(unsigned i = 0; i < M; i++) {
          const ull t1 = x1 - z1, t2 = x2 - z2;
          z1 = m.fma(z1, z1, C1), z2 = m.fma(z2, z2, C2);
          q1 = m.mul(q1, t1), q2 = m.mul(q2, t2);
        }
        q1 = m.mul(q1, x1 - z1), q2 = m.mul(q2, x2 - z2);
        const ull q3 = m.mul(q1, q2), g3 = gcd(n, q3);
        if(g3 == 1) { continue; }
        if(g3 != n) { return g3; }
        const ull g1 = gcd(n, q1), g2 = gcd(n, q2);
        const ull C = g1 != 1 ? C1 : C2, x = g1 != 1 ? x1 : x2;
        ull z = g1 != 1 ? y1 : y2, g = g1 != 1 ? g1 : g2;
        if(g == n) {
          do {
            z = m.fma(z, z, C);
            g = gcd(n, x - z);
          } while(g == 1);
        }
        if(g != n) { return g; }
        Z1 += 2, Z2 += 2;
        goto retry;
      }
    }
  }
  vector<ull> PrimeFactorize(ull n) {
    vector<ull> r;
    auto rec = [&](auto &&rec, ull n, vector<ull> &r) -> void {
      if(n <= 1) { return; }
      if(is_prime(n)) {
        r.emplace_back(n);
        return;
      }
      const ull p = PollardRho(n);
      rec(rec, p, r);
      rec(rec, n / p, r);
    };
    rec(rec, n, r);
    sort(r.begin(), r.end());
    return r;
  }
  vector<pair<ll, ll>> Prime(ll n) {
      auto ans = PrimeFactorize(n);
      vector<pair<ll, ll>> cur;
      for(ll i = 0; i < ans.size(); ++i){
          ll e = 1;
```

```
            while(i + 1 < ans.size() && ans[i + 1] == ans[i]) ++e, ++i;
            cur.push_back({ans[i], e});
        }
        return cur;
    }
    // auto get_tot = [](auto &&get_tot, vector<pair<ll, ll>>& prime, vector<ll>& tot, int pos, ll val){
    //     if(pos == prime.size()){
    //         tot.push_back(val);
    //         return;
    //     }
    //     for(ll j = 0, sum = 1; j <= prime[pos].second; ++j, sum *= prime[pos].first){
    //         get_tot(get_tot, prime, tot, pos + 1, val * sum);
    //     }
    // };
#endif
```