Template

Pray Never

2024 年 11 月 21 日

目录

1 data structure

1.1 trie.cpp

```
#ifdef IGNORE_THIS_FILE
  int cnt = 0;
  vector<vector<int> > node(1, vector<int>(26));
  vector<int> isval(1);
  auto insert = [&](string& s, int len) -> void {
    int p = 0;
    for(int i = 0; i < len; i++){
      int c = s[i] - 'a';
      if(!node[p][c]){
        node.push_back(vector<int>(26, 0));
        isval.push_back(0);
        node[p][c] = ++cnt;
      }
      p = node[p][c];
    }
    ++isval[p];
  };
  auto find = [&](string& s, int len) -> int {
    int p = 0;
    for(int i = 0; i < len; i++){
      int c = s[i] - 'a';
      if(!node[p][c]) return 0;
      p = node[p][c];
    }
    return isval[p];
  };
#endif
```

2   geometry

2.1   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
```

3 graph

3.1 LCA.cpp

```
1   #ifdef IGNORE_THIS_FILE
2     vector<vector<int> > a(n + 1, vector<int>(20)), v(n + 1);
3     vector<int> dep(n + 1);
4     auto build = [&](int u, int fa, auto&& self) -> void {
5       dep[u] = dep[fa] + 1, a[u][0] = fa;
6       for(int i = 1; i <= 19; ++i)
7         a[u][i] = a[a[u][i - 1]][i - 1];
8       for(int i: v[u]){
9         if(i == fa) continue;
10        self(i, u, self);
11      }
12    };
13    auto lca = [&](int x, int y) -> int {
14      if(dep[y] > dep[x]) swap(x, y);
15      for(int i = 19; i >= 0; --i){
16        if(dep[a[x][i]] >= dep[y])
17          x = a[x][i];
18      }
19      if(x == y) return x;
20      for(int i = 19; i >= 0; --i){
21        if(a[x][i] != a[y][i])
22          x = a[x][i], y = a[y][i];
23      }
24      return a[x][0];
25    };
26  #endif
27
28
29
```

3.2 tarjan.cpp

```
1   #ifdef IGNORE_THIS_FILE
2     int dfn_cnt = 0, scc_cnt = 0;
3     vector<int> dfn(n + 1), low(n + 1), scc_id(n + 1);
4     stack<int> s;
5     vector<bool> in_stack(n + 1);
6     vector<vector<int> > v(n + 1), scc(n + 1);
7     // 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
8     auto tarjan = [&](int u, auto&& self) -> void {
9       dfn[u] = low[u] = ++dfn_cnt;
10      s.push(u), in_stack[u] = true;
11      for(int i: v[u]){
12        if(!dfn[i]){
13          self(i, self);
14          low[u] = min(low[u], low[i]);
```

```
15          }
16        else if(in_stack[i])
17            low[u] = min(low[u], dfn[i]);
18      }
19    if(dfn[u] == low[u]){
20      int tp;
21      ++scc_cnt;
22      do{
23        tp = s.top();
24        scc_id[tp] = scc_cnt;
25        scc[scc_cnt].push_back(tp);
26        in_stack[tp] = false;
27        s.pop();
28      } while(tp != u);
29    }
30  };
31 #endif
```

4   heading

4.1   debug.h

```
1    #include <bits/stdc++.h>
2    #define typet typename T
3    #define typeu typename U
4    #define types typename... Ts
5    #define tempt template <typet>
6    #define tempu template <typeu>
7    #define temps template <types>
8    #define tandu template <typet, typeu>
9
10   tandu std::ostream& operator<<(std::ostream& os, const std::pair<T, U>& p) {
11   return os << '<' << p.ff << ',' << p.ss << '>';
12   }
13   template <
14   typet, typename = decltype(std::begin(std::declval<T>())),
15   typename = std::enable_if_t<!std::is_same_v<T, std::string>>>
16   std::ostream& operator<<(std::ostream& os, const T& c) {
17   auto it = std::begin(c);
18   if (it == std::end(c)) return os << "{}";
19   for (os << '{' << *it; ++it != std::end(c); os << ',' << *it);
20   return os << '}';
21   }
22   #define debug(arg...) \
23   do { \
24    std::cerr << "[" #arg "] :"; \
25    dbg(arg); \
26   }while(false)
27
28   temps void dbg(Ts... args) {
29   (..., (std::cerr << ' ' << args));
30   std::cerr << '\n';
31   }
32
```

4.2   duipai.cpp

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

```
12        return 0;
13      }
14      std::cerr << "AC\n";
15    }
16  #endif
```

4.3   heading.cpp

```
1   #include <bits/stdc++.h>
2   using namespace std;
3   using ll = long long;
4   using i128 = __int128;
5   #define ff first
6   #define ss second
7   #include "debug.h"
8   constexpr int mod = 998244353;
9   constexpr ll INF = 1e18;
10  constexpr double pi = 3.141592653589793;
11  constexpr double eps = 1e-6;
12
13
```

5   math

5.1   Eratost.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  vector<int> sieve(int n){
    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
```

5.2   Euler.cpp

```cpp
#ifdef IGNORE_THIS_FILE
  // vector<int> fac(n + 1);
  vector<int> sieve(int n){
    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
```

5.3 pollarrho.cpp

```cpp
1   #ifdef IGNORE_THIS_FILE
2     using ll = long long;
3     using ull = unsigned long long;
4     bool is_prime(ull n) {
5       if(n == 2) { return true; }
6       if(n % 2 == 0) { return false; }
7       auto internal_pow = [&](ull x, ull y) {
8         ull r = 1;
9         __uint128_t c = x;
10        for(; y; y >>= 1, c = c * c % n) {
11          if(y & 1) { r = __uint128_t(r) * c % n; }
12        }
13        return r;
14      };
15      auto MillerRabin = [&](ull a) {
16        if(n <= a) { return true; }
17        int e = __builtin_ctzll(n - 1);
18        ull z = internal_pow(a, (n - 1) >> e);
19        if(z == 1 || z == n - 1) { return true; }
20        while(--e) {
21          z = __uint128_t(z) * z % n;
22          if(z == 1) { return false; }
23          if(z == n - 1) { return true; }
24        }
25        return false;
26      };
27      vector<ull> cur;
28      if(n < 4759123141) cur = vector<ull>{2, 7, 61};
29      else cur = vector<ull>{2, 325, 9375, 28178, 450775, 9780504, 1795265022};
30      return all_of(cur.begin(), cur.end(), [&](auto x) { return MillerRabin(x); });
31    }
32
33    struct Montgomery {
34      ull mod, R;
35      public:
36      Montgomery(ull n): mod(n), R(n) {
37        for(int i = 0; i < 5; i++) { R *= 2 - mod * R; }
38      }
39      ull fma(ull a, ull b, ull c) const {
40        const __uint128_t d = __uint128_t(a) * b;
41        const ull e = c + mod + (d >> 64);
42        const ull f = ull(d) * R;
43        const ull g = (__uint128_t(f) * mod) >> 64;
44        return e - g;
45      }
46      ull mul(ull a, ull b) const { return fma(a, b, 0); }
47    };
48    ull PollardRho(ull n) {
49      if(n % 2 == 0) { return 2; }
50      const Montgomery m(n);
51      constexpr ull C1 = 1, C2 = 2, M = 512;
```

```cpp
      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;
```

```
106            while(i + 1 < ans.size() && ans[i + 1] == ans[i]) ++e, ++i;
107            cur.push_back({ans[i], e});
108        }
109        return cur;
110    }
111    // auto get_tot = [](auto &&get_tot, vector<pair<ll, ll>>& prime, vector<ll>& tot, int pos, ll val){
112    //    if(pos == prime.size()){
113    //        tot.push_back(val);
114    //        return;
115    //    }
116    //    for(ll j = 0, sum = 1; j <= prime[pos].second; ++j, sum *= prime[pos].first){
117    //        get_tot(get_tot, prime, tot, pos + 1, val * sum);
118    //    }
119    // };
120    #endif
```

6 sort

6.1 mergesort.cpp

```cpp
#ifdef IGNORE_THIS_TILE
  void merge_sort(int l, int r, vector<int>& a) {
    if(l == r)
      return;
    int mid = (l + r) >> 1;
    merge_sort(l, mid, a);
    merge_sort(mid + 1, r, a);
    auto merge = [](int l, int r, int mid, vector<int>& a) -> void {
      vector<int> b;
      int lp = l;
      int rp = r + 1;
      while(lp <= mid && rp <= r){
        if(a[lp] < a[rp])
          b.push_back(a[lp++]);
        else
          b.push_back(a[rp++]);
      }
      while(lp <= mid)
        b.push_back(a[lp++]);
      while(rp <= r)
        b.push_back(a[rp++]);
      copy(b.begin(), b.end(), a.begin() + l);
    };
    merge(l, r, mid, a);
  };
#endif
```