Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0398 – Романовский Владислав Александрович

A	B	C	D	E	F	sum
100	100	100	100	40	98	538

Task A

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#ifndef LOCAL
#define endl '\n'
#endif
const int N = 200;
int dp[N];
signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    freopen("test.txt", "r", stdin);
#endif
    int a, b, n;
    cin >> a >> b >> n;
    int ans = 0;
    for (int k = 0; k < max(a, b); k++) {
        // if(k == a)continue;
        // if(k == b)continue;
        for(int i = 0; i < N; i++) dp[i] = 0;
        dp[k] = 1;
        for (int i = k; i < n; i++) {
```

```
            dp[i + a] += dp[i];
            dp[i + b] += dp[i];
        }
        if (b > k && (b - k) <= n) {
            ans += dp[n - (b - k)];
        }
        if (a > k && (a - k) <= n) {
            ans += dp[n - (a - k)];
        }
    }
    cout << ans << endl;
}
```

Task B

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#ifndef LOCAL
#define endl '\n'
#endif

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    freopen("test.txt", "r", stdin);
#endif
    int n;
    cin >> n;
    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    auto dist = [&](int a, int b, int c, int x, int y, int z) { return (
   abs(a - x) + abs(b - y) + abs(c - z)) / 2; };

    vector<int> a(3);
    cin >> a[0] >> a[1] >> a[2];
    int ans = 1e18;
    ans = dist(x1, y1, z1, a[0], a[1], a[2]);
    vector<int> p(3);
    iota(all(p),0ll);
    vector<int> b = a;
    do {
        a[p[0]] += 2 * n;
        a[p[1]] += n;
        a[p[2]] += n;
        ans = min(ans, dist(x1, y1, z1, a[0], a[1], a[2]));
```

```
        a = b;
        a[p[0]] -= 2 * n;
        a[p[1]] += n;
        a[p[2]] += n;
        ans = min(ans, dist(x1, y1, z1, a[0], a[1], a[2]));
        a = b;
        a[p[0]] += 2 * n;
        a[p[1]] -= n;
        a[p[2]] -= n;
        ans = min(ans, dist(x1, y1, z1, a[0], a[1], a[2]));
        a = b;
        a[p[0]] -= 2 * n;
        a[p[1]] -= n;
        a[p[2]] -= n;
        ans = min(ans, dist(x1, y1, z1, a[0], a[1], a[2]));
        a = b;

    } while (next_permutation(all(p)));
    cout << ans << endl;
}
```

Task C

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#ifndef LOCAL
#define endl '\n'
#endif
struct sufmass {
    string s;
    vector<int> P;
    vector<int> lcp;
    void SORT(vector<int>& p, vector<int>& c) {
        int n = c.size();
        vector<int> cnt(n), pos(n);
        for (auto x : c)
            cnt[x]++;
        for (int i = 1; i < n; i++)
            pos[i] = pos[i - 1] + cnt[i - 1];
        vector<int> np(n);
        for (auto i : p) {
            auto& j = pos[c[i]];
            np[j++] = i;
        }
        p = np;
    }
```

```
void get_lcp() {
    int n = s.size();
    lcp.assign(n, {});
    vector<int> pos(n);
    for (int i = 0; i < n; ++i) {
        pos[P[i]] = i;
    }
    int l = 0;
    for (int ii = 0; ii < n - 1; ++ii) {
        if (pos[ii] == 0) {
            l--;
            l = max<int>(0, l);
            continue;
        }
        int I = ii;
        int J = P[pos[ii] - 1];
        while (s[(I + l) % n] == s[(J + l) % n]) {
            l++;
        }
        lcp[pos[ii]] = l;
        l--;
        l = max<int>(0, l);
    }
}
void build(string S) {
    s = S;
    s += "$";
    int n = s.size();
    vector<pair<int, int>> sp;
    for (int i = 0; i < n; i++)
        sp.push_back({s[i] - 'a', i});
    sort(all(sp));
    vector<int> p(n), c(n);
    for (int i = 0; i < n; i++)
        p[i] = sp[i].second;
    for (int i = 1; i < n; i++)
        c[p[i]] = c[p[i - 1]] + (sp[i].first != sp[i - 1].first);
    for (int k = 1; k < n; k <<= 1) {
        for (int i = 0; i < n; i++)
            p[i] = (p[i] - k + n) % n;
        SORT(p, c);
        vector<int> C(n);
        pair<int, int> prev = {c[p[0]], c[(p[0] + k) % n]};
        for (int i = 1; i < n; i++) {
            pair<int, int> cur = {c[p[i]], c[(p[i] + k) % n]};
            C[p[i]] = C[p[i - 1]] + (cur != prev);
            prev = cur;
        }
        c = C;
    }
    P = p;
    get_lcp();
}
```

```
};
const int mod=(int)1e9+ 7;
string get_smallest(string s) {
    s = s + s;
    sufmass S;
    S.build(s);
    int fst = S.P[s.size()];
    string ans;
    for(int j = fst, i = 0; i < s.size() / 2; i++,j++,j %= s.size()) {
        ans += (char)s[j];
    }
    return ans;
}
int mul(int a, int b) {
    a = (__int128)a * b % mod;
    return a;
}
int add(int a, int b) {
    return (a + b) % mod;
}
int get_hesh(string s) {
    int ans = 0;
    int cur = 131239;
    for(int i = 0; i < s.size(); i++) {
        int c =(s[i]-'0') + 1;
        ans = mul(ans, cur);
        ans = add(ans, c);
    }
    return ans;
}
signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    freopen("test.txt", "r", stdin);
#endif
    string s;
    cin >> s;
    if(s == "first") {
        int n;
        cin >> n;
        cin >> s;
        string S = get_smallest(s);
        int hesh = get_hesh(S);
        cout << hesh << endl;
    } else {
        int n;
        cin >> n;
        cin >> s;
        int hesh1;
        cin >> hesh1;
        string S = get_smallest(s);
        int hesh = get_hesh(S);
```

```
        if(hesh1 == hesh) {
            cout << "YES" << endl;
        } else {
            cout << "NO" << endl;
        }

    }
}
```

Task D

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
const int N = 510;
int dp[N][N];
vector<int> P;
signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    {
        int n;
        cin >> n;
        for (int i = 0; i < n; i++) {
            int x;
            cin >> x;
            P.push_back(x);
        }
    }
#endif
    int n;
    cin >> n;

    int col = n * (n - 1);
    vector<int> ANS(n);
    for (int i = 0; i < n; i++) {
        ANS[i] = n - i - 1;
    }
    auto SWAP = [&](int a, int b, vector<int> &p) { swap(p[a], p[b]); };
    auto shift = [&](vector<int> &p) { rotate(p.begin(), p.begin() + 1, p
   .end()); };
    vector<int> p(n);
    iota(all(p), 0ll);
    auto go = [&](int a, int b) {
        cout << a + 1 << " " << b + 1 << endl;
        SWAP(a, b, p);
```

```
        shift(p);
#ifdef LOCAL
        vector<int> rem = P;
        SWAP(a, b, P);
        shift(P);
        if (rem == ANS)
            return '!';
        if (rem < P)
            return '<';
        else if (rem > P)
            return '>';
#else
        char x;
        cin >> x;
        return x;
#endif
    };
    while (col > 0) {
        int cur = p[0];
        bool ok = false;
        for (int i = 0; i < n; i++) {
            if (p[0] == i)
                continue;
            if (dp[p[0]][i] == 0) {
                ok = true;

                int id = 0;
                for (int j = 0; j < n; j++) {
                    if (p[j] == i) {
                        id = j;
                        break;
                    }
                }
                char x = go(1, id);
                if (x == '!')
                    return 0;
                if (x == '<') {
                    dp[cur][i] = 1;
                    dp[i][cur] = -1;
                    col-=2;
                } else {
                    dp[cur][i] = -1;
                    dp[i][cur] = 1;
                    col-=2;
                }
                break;
            }
        }
        if (!ok) {
            char x = go(0, 0);
            if (x == '!')
                return 0;
        }
```

```
    }
    vector<int> who(n);
    for (int i = 0; i < n; i++) {
        int col = 0;
        for (int j = 0; j < n; j++) {
            if (dp[i][j] == 1)
                col++;
        }
        who[i] = col;
    }
    for (int i = n - 1; i >= 0; i--) {
        int cur = 0;
        for (int j = 0; j < n; j++) {
            if (who[j] == i)
                cur = j;
        }
        int id = 0;
        for (int j = 0; j < n; j++) {
            if (p[j] == cur) {
                id = j;
                break;
            }
        }
        char x = go(0, id);
        if (x == '!')
            return 0;
    }
}
```

Task E

```
#include <bits/stdc++.h>
using namespace std;

using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#ifndef LOCAL
#define endl '\n'
#endif
const int N = 1e7 + 100;
int col[N];
vector<int> d;
vector<int> G[N];
void dfs(int v) {
    d.push_back(v);
    if(d.size() > N)return;
    col[v]++;
    col[v] %= G[v].size();
    dfs(G[v][(col[v] - 1 + G[v].size()) % G[v].size()]);
```

```
}
vector<int> z_func(vector<int> s) {
  vector<int> Z(s.size(),0);
  int n = s.size();
  int l = 0;
  int r = 0;
  for (int i = 1; i < n; i++)
  {
    if (i <= r)
    {
      Z[i] = min(Z[i - l], r - i);
    }
    while (i + Z[i] < n && s[Z[i] + i] == s[Z[i]])
    {
      Z[i] ++;
    }
    if (i + Z[i] > r)
    {
      r = i + Z[i];
      l = i;
    }
  }
  return Z;
}

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    freopen("test.txt", "r", stdin);
#endif
    int n,q;
    cin >> n >> q;
    for(int i = 0; i < n; i++) {
        int d;
        cin >> d;
        for(int  j = 0; j < d; j++) {
            int x;
            cin >> x;
            x--;
            G[i].push_back(x);
        }
    }

    dfs(0);
    int fst = 2e6 + 100;
    vector<int> dn;
    for(int i = fst; i < N; i++) {
        dn.push_back(d[i]);
    }
    int len;
    vector<int> z = z_func(dn);
    for(int i = 0; i < z.size(); i++) {
```

```
        if(i + z[i] == dn.size()) {
            len = i;
            break;
        }
    }
    for(int i = 0; i < q; i++) {
        int x;
        cin >> x;
        if(x < fst) {
            cout << d[x] + 1 << " ";
        } else {
            x -= fst;
            x %= len;
            cout << dn[x] + 1 << " ";
        }
    }
}
```

Task F

```
#include <bits/stdc++.h>
using namespace std;
#pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx,avx2")
using ll = long long;
#define int long long
#define all(a) (a).begin(), (a).end()
#define rall(a) (a).rbegin(), (a).rend()
#define gen(a, b) uniform_int_distribution<int>(a, b)(rnd)
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());
#ifndef LOCAL
#define endl '\n'
#endif
double t = 1000;
const int INF = LONG_LONG_MAX;
signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
#ifdef LOCAL
    freopen("test.txt", "r", stdin);
#endif
    int n, m;
    cin >> n >> m;
    vector<vector<char>> cur(n, vector<char>(m, '.'));
    vector<vector<int>> c(n, vector<int>(m));
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            char x;
            cin >> x;
            int k = (x - '0');
            c[i][j] = k;
        }
    }
```

```
int ans = 0;
for (int i = 0; i < n; i++) {
    for (int j = 0; j < m; j += 2) {
        cur[i][j] = 'L';
        cur[i][j + 1] = 'R';
        ans += c[i][j] * c[i][j + 1];
    }
}
auto randD = [&]() { return (double)gen(0, INF) / INF; };
vector<vector<char>> ANS = cur;
vector<array<int, 3>> reve;
auto rev = [&](char x) {
    if (x == 'T')
        return 'B';
    if (x == 'B')
        return 'T';
    if (x == 'L')
        return 'R';
    return 'L';
};
auto del = [&](int x, int y) {
    if (cur[x][y] == '.')
        return;
    int nx;
    int ny;
    if (cur[x][y] == 'T') {
        nx = (x + 1) % n;
        ny = y;
    } else if (cur[x][y] == 'B') {
        nx = (x - 1 + n) % n;
        ny = y;
    } else if (cur[x][y] == 'L') {
        nx = x;
        ny = (y + 1) % m;
    } else {
        nx = x;
        ny = (y - 1 + m) % m;
    }
    ans -= c[nx][ny] * c[x][y];
    reve.push_back({x, y, cur[x][y]});
    reve.push_back({nx, ny, cur[nx][ny]});
    cur[x][y] = '.';
    cur[nx][ny] = '.';
};
auto add = [&](int x, int y) {
    if (cur[x][y] == '.')
        return;
    int nx;
    int ny;
    if (cur[x][y] == 'T') {
        nx = (x + 1) % n;
        ny = y;
    } else if (cur[x][y] == 'B') {
```

```
            nx = (x - 1 + n) % n;
            ny = y;
        } else if (cur[x][y] == 'L') {
            nx = x;
            ny = (y + 1) % m;
        } else {
            nx = x;
            ny = (y - 1 + m) % m;
        }
        if (cur[nx][ny] != '.')
            del(nx, ny);
        ans += c[nx][ny] * c[x][y];
        reve.push_back({nx, ny, cur[nx][ny]});
        cur[nx][ny] = rev(cur[x][y]);
    };
    vector<char> choose = {'L', 'R', 'T', 'B'};
    while (true) {
        t *= 0.999999;
        if (clock() * 1.0 / CLOCKS_PER_SEC > 1.98)
            break;
        int x = gen(0, n - 1);
        int y = gen(0, m - 1);
        int curans = ans;
        del(x, y);
        int vec = gen(0, 3);
        reve.push_back({x, y, cur[x][y]});
        cur[x][y] = choose[vec];
        add(x, y);
        if (ans > curans || randD() < exp((double)(ans - curans) / t)) {
            reve.clear();
        } else {
            while (!reve.empty()) {
                cur[reve.back()[0]][reve.back()[1]] = reve.back()[2];
                reve.pop_back();
            }
            ans = curans;
        }
    }
    cout << ans << endl;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cout << cur[i][j];
        }
        cout << endl;
    }
}
```