Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0233 – Кочнев Елисей Владиславович

A	B	C	D	E	F	sum
100	100	40	75	20	47	382

Task A

```
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
typedef long double ld;

const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
         __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/
```

```
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
    int a, b, n; cin >> a >> b >> n;
    vector<ll> cnt(n + 1);
    ll ans = 0;
    cnt[0] = 1;
    for (int i = 0; i <= n; ++i) {
        if (i + a < n) {
            cnt[i + a] += cnt[i];
        }
        if (i + b < n) {
            cnt[i + b] += cnt[i];
        }
        if (i + a == n) {
            ans += a * cnt[i];
        }
        if (i + b == n) {
            ans += b * cnt[i];
        }
    }
    cout << ans;
    return 0;
}
```

Task B

```
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
typedef long double ld;

const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
        __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
```

```
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/


signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
    ll n; cin >> n;
    ll x1, y1, z1; cin >> x1 >> y1 >> z1;
    ll x2, y2, z2; cin >> x2 >> y2 >> z2;
    ll mi = (abs(x1 - x2) + abs(y1 - y2) + abs(z1 - z2)) / 2;
    if (x2 < 0) {
        mi = min(mi, (abs(x1 - x2 - 2 * n) + abs(y1 - y2 + n) + abs(z1 -
   z2 + n)) / 2ll);
    } else {
        mi = min(mi, (abs(x1 - x2 + 2 * n) + abs(y1 - y2 - n) + abs(z1 -
   z2 - n)) / 2ll);
    }
    if (y2 < 0) {
        mi = min(mi, (abs(x1 - x2 + n) + abs(y1 - y2 - 2 * n) + abs(z1 -
   z2 + n)) / 2ll);
    } else {
        mi = min(mi, (abs(x1 - x2 - n) + abs(y1 - y2 + 2 * n) + abs(z1 -
   z2 - n)) / 2ll);
    }
    if (z2 < 0) {
        mi = min(mi, (abs(x1 - x2 + n) + abs(y1 - y2 + n) + abs(z1 - z2 -
     2 * n)) / 2ll);
    } else {
        mi = min(mi, (abs(x1 - x2 - n) + abs(y1 - y2 - n) + abs(z1 - z2 +
     2 * n)) / 2ll);
    }
    cout << mi;
    return 0;
}
```

Task C

```
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
typedef long double ld;
```

```
const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
         __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/

int a, b;
mt19937 rnd(566);
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
    string t, s; cin >> t;
    int n; cin >> n;
    if (n <= 100) {
        a = b = 1;
    } else {
        a = n / 90 + rnd() % 100;
        b = n / 90 + rnd() % 100;
    }
    string q;
    if (t == "first") {
        cin >> s;
        int i = 0;
        int k = 0;
        while(i < n - 10) {
            q += s[i];
            if (!k) {
                i += a;
            } else {
                i += b;
            }
            k = (k + 1) % 2;
        }
        for (int i = max(n - 10, 0); i < n; ++i) {
            q += s[i];
```

```
        }
        cout << q << endl;
    } else {
        cin >> s;
        cin >> q;
        s = s + s;
        int j, u;
        int k;
        bool f;
        for (int i = 0; i < n; ++i) {
            j = 0;
            k = 0;
            u = 0;
            f = true;
            while(j < n - 10) {
                if (s[i + j] != q[u]) {
                    f = false;
                    break;
                }
                if (!k) {
                    j += a;
                } else {
                    j += b;
                }
                k = (k + 1) % 2;
                u++;
            }
            u = 0;
            for (j = max(n - 10, 0); j < n; ++j) {
                if (s[i + j] != q[max((int) q.size() - 10, 0) + u]) {
                    f = false;
                    break;
                }
                u++;
            }
            if (f) {
                cout << "YES";
                exit(0);
            }
        }
        cout << "NO";
    }
    return 0;
}
```

Task D

```
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
```

```
typedef long double ld;

const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
         __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
    int n; cin >> n;
    char ans = '=';
    cout << "1 1" << endl;
    while(ans != '!') {
        bool f = true;
        for (int i = 0; i < n; ++i) {
            cin >> ans;
            if (ans == '!') {
                break;
            }
            if (ans == '>' && f) {
                cout << "1 " << n << endl;
                f = false;
            } else {
                cout << "1 1" << endl;
            }
        }
    }
    return 0;
}
```

Task E

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

```
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
typedef long double ld;

const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
         __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/

vector<deque<int>> g;
vector<pair<ll, ll>> t;
vector<int> ans;
int timer = 0;
int u = 0;
void dfs(int v) {
    if (u >= (int) ans.size()) {
        return;
    }
    if (timer == t[u].first) {
        ans[t[u].second] = v + 1;
        u++;
    }
    timer++;
    g[v].push_back(g[v].front());
    g[v].pop_front();
    dfs(g[v].back());
}

signed main() {
    // ios_base::sync_with_stdio(false);
    // cin.tie(nullptr);
    // cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
```

```
    int n, q; cin >> n >> q;
    g.resize(n);
    int k, v;
    for (int i = 0; i < n; ++i) {
        cin >> k;
        while(k--) {
            cin >> v;
            v--;
            g[i].push_back(v);
        }
    }
    t.resize(q);
    ans.resize(q);
    for (int i = 0; i < q; ++i) {
        cin >> t[i].first, t[i].second = i;
    }
    sort(t.begin(), t.end());
    dfs(0);
    for (auto & i : ans) cout << i << ' ';
    return 0;
}
```

Task F

```
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>

using namespace std;

typedef long long ll;
typedef long double ld;

const int inf = 2e9;
const ll mod = 1e9 + 31;
const ll p = 31;
/*
using indexed_set = __gnu_pbds::tree<pair<ll,int>, __gnu_pbds::null_type,
     less<>, __gnu_pbds::rb_tree_tag,
        __gnu_pbds::tree_order_statistics_node_update>;
*/

/*
long long get(int r) {
    long long sum = 0;
    for (; r >= 0; r = (r & (r + 1)) - 1) {
        sum += tree[r];
    }
    return sum;
}
void upd(int i, int x) {
    for (; i < n; i = (i | (i + 1))) tree[i] += x;
}*/
```

```
signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
    // freopen("battle.in", "r", stdin);
    // freopen("battle.out", "w", stdout);
    int w, h; cin >> w >> h;
    vector<string> a(w);
    for (auto &i : a) cin >> i;
    ll sum = 0;
    for (int i = 0; i < w; ++i) {
        for (int j = 1; j < h; j += 2) {
            sum += (a[i][j - 1] - '0') * (a[i][j] - '0');
        }
        if (h % 2 == 1 && i % 2 == 0 && i != w - 1) {
            sum += (a[i][h - 1] - '0') * (a[i + 1][h - 1] - '0');
        }
    }
    cout << sum << '\n';
    for (int i = 0; i < w; ++i) {
        for (int j = 1; j < h; j += 2) {
            cout << "LR";
        }
        if (h % 2 == 1 && i % 2 == 0 && i != w - 1) {
            cout << 'T';
        } else if (h % 2 == 1 && i % 2 == 1) {
            cout << 'B';
        }
        if(i != w - 1) cout << '\n';
    }
    if (h % 2 == 1 && w % 2 == 1) {
        cout << '.';
    }
    return 0;
}
```