Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0140 – Доронин Фёдор Александрович

A	B	C	D	E	F	sum
100	100	100	35	40	91	466

Task A

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <map>
#include <random>
#include <ctime>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;
#define int int64_t
typedef tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;
const int INF = 1e18 + 117, MAXN = 1e6 + 117, M = 1e9 + 7;
mt19937 gen(time(0));

int a, b, n;
int pred = -1, first = -1, ans = 0;

bool check(int x) {
    for (int i = 0; i * b <= n; ++i) {
        if ((n - i*b) % a == 0) return true;
    }
    return false;
}

vector<int> dp(100, -1);
```

```
int f(int ind) {
    if (ind >= n) return 0;
    if (!check(n - ind + first)) return 0;
    if (dp[ind] != -1) return dp[ind];
    int cur = 0;
    if (n - ind + first == a || n - ind + first == b) {
        cur = 1;
    }
    dp[ind] = cur + f(ind + a) + f(ind + b);
    return dp[ind];
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
#ifdef FEDORODORO
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    cin >> a >> b >> n;
    /*int ans = 0;
    for (int fs = 0; fs < 100; ++fs) {
        for (int sc = 0; sc < 100; ++sc) {
            if (fs == 0 && sc == 0) continue;
            if (n != fs*a + sc*b) continue;
            if (fs == 0) ans += sc;
            else if (sc == 0) ans += fs;

        }
    }*/
    int res = 0;
    for (int i = 0; i < n; ++i) {
        first = i;
        res += f(i);
        for (auto& e : dp) e = -1;
    }
    cout << res << '\n';
    return 0;
}
```

Task B

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <map>
#include <random>
#include <ctime>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
```

```
using namespace __gnu_pbds;
#define int int64_t
typedef tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;
const int INF = 1e18 + 117, MAXN = 1e6 + 117, M = 1e9 + 7;
mt19937 gen(time(0));

int gt(int x1, int y1, int z1, int x2, int y2, int z2) {
    int ans = INF;
    ans = min(ans, abs(x1 - x2) + abs(y1 - y2));
    ans = min(ans, abs(x1 - x2) + abs(z1 - z2));
    ans = min(ans, abs(y1 - y2) + abs(z1 - z2));
    return ans;
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
#ifdef FEDORODORO
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n, x1, y1, z1, x2, y2, z2;
    cin >> n >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
    int ans = INF;
    ans = min(ans, gt(x1, y1, z1, x2, y2, z2));
    ans = min(ans, gt(x1, y1, z1, x2 - n, y2 - n, z2 - 2*n));
    ans = min(ans, gt(x1, y1, z1, x2 - n, y2 - 2*n, z2 - n));
    ans = min(ans, gt(x1, y1, z1, x2 - 2*n, y2 - n, z2 - n));

    ans = min(ans, gt(x1, y1, z1, x2 + n, y2 + n, z2 + 2*n));
    ans = min(ans, gt(x1, y1, z1, x2 + n, y2 + 2*n, z2 + n));
    ans = min(ans, gt(x1, y1, z1, x2 + 2*n, y2 + n, z2 + n));

    ans = min(ans, gt(x1, y1, z1, x2 + n, y2 + n, z2 - 2*n));
    ans = min(ans, gt(x1, y1, z1, x2 + n, y2 - 2*n, z2 + n));
    ans = min(ans, gt(x1, y1, z1, x2 - 2*n, y2 + n, z2 + n));


    ans = min(ans, gt(x1, y1, z1, x2 - n, y2 - n, z2 + 2*n));
    ans = min(ans, gt(x1, y1, z1, x2 - n, y2 + 2*n, z2 - n));
    ans = min(ans, gt(x1, y1, z1, x2 + 2*n, y2 - n, z2 - n));

    cout << ans << '\n';
    return 0;
}
```

Task C

```
M = 10**98 + 117
p = 2
MAXN = 2*(10**5) + 117
```

```python
ps = [0] * MAXN
ps[0] = 1
for i in range(1, MAXN):
    ps[i] = ps[i - 1] * p % M
tp = input()
if tp == "first":
    n = int(input())
    s = input()
    h = 0
    for i in range(n):
        if s[i] == '1':
            h = h * p + 2
            h %= M
        else:
            h = h * p + 1
            h %= M
    print(h)
else:
    n = int(input())
    s = input()
    h = int(input())
    s += s
    hs = [0] * (len(s) + 1)
    PS = ps[n]
    for i in range(len(s)):
        if s[i] == '1':
            hs[i + 1] = (hs[i] * p + 2) % M
        else:
            hs[i + 1] = (hs[i] * p + 1) % M
    for i in range(n):
        cur_h = (hs[i + n] - hs[i] * PS % M + M) % M
        if cur_h == h:
            print("YES")
            quit()
    print("NO")
```

Task D

```cpp
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <map>
#include <random>
#include <ctime>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;
#define int int64_t
typedef tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;
const int INF = 1e18 + 117, MAXN = 1e6 + 117, M = 1e9 + 7;
```

```
mt19937 gen(time(0));



signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    int n;
    cin >> n;
    if (n < 7) {
        while (true) {
            cout << 1 << ' ' << gen() % n + 1 << endl;
            cout.flush();
            char ans; cin >> ans;
            if (ans == '!') break;
        }
        return 0;
    }
    while (true) {
        cout << 1 << ' ' << gen() % n + 1 << endl;
        cout.flush();
        char ans; cin >> ans;
        if (ans == '!') break;
    }
    return 0;
}
//
```

Task E

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <map>
#include <random>
#include <ctime>
#include <deque>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;
#define int int64_t
typedef tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;
const int INF = 1e18 + 117, MAXN = 3e6, M = 1e9 + 7;
mt19937 gen(time(0));

int n, q;
vector<deque<int>> g;
vector<int> ts;
vector<int> way;
```

```
void dfs(int v) {
    if (way.size() > MAXN) return;
    way.push_back(v);
    while (true) {
        int tmp = g[v].front();
        g[v].push_back(g[v].front());
        g[v].pop_front();
        dfs(tmp);
        if (way.size() > MAXN) return;
    }
}

bool check(int fs, int sc) {
    for (int i = 0; i < 2*n - 2; ++i) {
        if (fs + i >= way.size() || sc + i >= way.size()) return false;
        if (way[fs + i] != way[sc + i]) return false;
    }
    return true;
}

signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
#ifdef FEDORODORO
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    cin >> n >> q;
    g.resize(n);
    for (int i = 0; i < n; ++i) {
        int k; cin >> k;
        while (k--) {
            int v; cin >> v; --v;
            g[i].push_back(v);
        }
    }
    dfs(0);
   // for (auto& e : way) cout << e << ' ';
    // cout << '\n';
    ts.resize(q);
    for (auto& e : ts) cin >> e;
    /*if (ts.back() < 2e5) {
        for (int i = 0; i < q; ++i) {
            // cin >> ts[i];
            if (ts[i] == 0) {
                cout << "1 ";
                continue;
            }
            cout << way[1 + (ts[i] - 1) % (2*n - 2)] + 1 << ' ';
        }
        return 0;
```

```
    }*/
    int S = 1;
    for (int st = 0; st < way.size(); ++st) {
        if (!check(st, st + 2*n - 2)) continue;
        S = st;
    }
    for (int i = 0; i < q; ++i) {
        if (ts[i] < S) {
            cout << way[ts[i]] + 1 << ' ';
            continue;
        }
        cout << way[S + (ts[i] - S) % (2*n - 2)] + 1 << ' ';
    }
    return 0;
}
//
```

Task F

```
#include <iostream>
#include <vector>
#include <set>
#include <algorithm>
#include <map>
#include <random>
#include <ctime>
#include <ext/pb_ds/assoc_container.hpp>

using namespace std;
using namespace __gnu_pbds;
#define int int64_t
typedef tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;
const int INF = 1e18 + 117, MAXN = 1e6 + 117, M = 1e9 + 7;
mt19937 gen(time(0));



signed main() {
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
#ifdef FEDORODORO
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int h, w; cin >> h >> w;
    vector<vector<int>> a(h, vector<int> (w));
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            char smb; cin >> smb;
            a[i][j] = smb - '0';
        }
```

```
}
vector<vector<char>> ans(h, vector<char> (w, '.'));
int cnt = 1500, res = 0;
while (cnt--) {
    int cur = -1, ii, jj, tmp;
    char tp;
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            if (ans[i][j] != '.') continue;
            // down
            if (ans[(i + 1)%h][j] == '.') {
                tmp = a[(i + 1)%h][j] * a[i][j];
                if (cur < tmp) {
                    cur = tmp;
                    tp = 'T';
                    ii = i; jj = j;
                }
            }
            // up
            if (ans[(i - 1 + h)%h][j] == '.') {
                tmp = a[(i - 1 + h)%h][j] * a[i][j];
                if (cur < tmp) {
                    cur = tmp;
                    tp = 'B';
                    ii = i; jj = j;
                }
            }
            // left
            if (ans[i][(j - 1 + w) % w] == '.') {
                tmp = a[i][(j - 1 + w) % w] * a[i][j];
                if (cur < tmp) {
                    cur = tmp;
                    tp = 'R';
                    ii = i; jj = j;
                }
            }
            // right
            if (ans[i][(j + 1) % w] == '.') {
                tmp = a[i][(j + 1) % w] * a[i][j];
                if (cur < tmp) {
                    cur = tmp;
                    tp = 'L';
                    ii = i; jj = j;
                }
            }
        }
    }
    if (cur == -1) break;
    res += cur;
    ans[ii][jj] = tp;
    if (tp == 'T') {
        ans[(ii + 1)%h][jj] = 'B';
    }
```

```
        if (tp == 'B') {
            ans[(ii - 1 + h)%h][jj] = 'T';
        }
        if (tp == 'R') {
            ans[ii][(jj - 1 + w) % w] = 'L';
        }
        if (tp == 'L') {
            ans[ii][(jj + 1) % w] = 'R';
        }
    }
    cout << res << '\n';
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) cout << ans[i][j];
        cout << '\n';
    }
    return 0;
}
// Designed by FEDORODORO
```