Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0372 – Полянский Владислав Николаевич

A	B	C	D	E	F	sum
100	100	100	-	40	47	387

Task A

```
def fact(n):
    res = 1
    for i in range(1, n + 1):
        res *= i
    return res


def cnk(n, k):
    if k < 0 or n < 0:
        return 0
    return fact(n) // fact(k) // fact(n - k)


a, b, n = map(int, input().split())
if a > b:
    a, b = b, a
ans = 0
for i in range(n//b + 1):
    j = (n - i * b) // a
    if (n - i * b) % a != 0:
        continue
    fir = cnk(i + j - 1, i - 1)
    sec = cnk(i + j - 1, i)
    if i == 0:
        fir = 0
    if j == 0:
        sec = 0
    ans += fir * b
    ans += sec * a
print(ans)
```

Task B

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>

using namespace std;
typedef long long ll;
const ll MOD = 1e9 + 7;

ll solve(ll n, ll a, ll b, ll c, ll d, ll e, ll f) {
  return max(max(abs(d - a), abs(e - b)), abs(f - c));
}

int main()
{
  cin.tie(0);
  ios_base::sync_with_stdio(false);
  ll n; cin >> n;
  ll a, b, c; cin >> a >> b >> c;
  ll d, e, f; cin >> d >> e >> f;
  ll ans = 1e12;
  ans = min(ans, solve(n, a, b, c, d, e, f));
  ans = min(ans, solve(n, a, b, c, d - n, e + 2 * n, f - n));
  ans = min(ans, solve(n, a, b, c, d + 2 * n, e - n, f - n));
  ans = min(ans, solve(n, a, b, c, d - n, e - n, f + 2 * n));
  ans = min(ans, solve(n, a, b, c, d + n, e - 2 * n, f + n));
  ans = min(ans, solve(n, a, b, c, d - 2 * n, e + n, f + n));
  ans = min(ans, solve(n, a, b, c, d + n, e + n, f - 2 * n));
  cout << ans << endl;
}
```

Task C

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>

using namespace std;
typedef long long ll;

const ll BASE1 = 5;
const ll BASE2 = 9;
const ll BASE3 = 17;

const ll MOD1 = 1e9 + 7;
const ll MOD2 = 1e12 + 13;
const ll MOD3 = 1e13 + 1;

struct StringHash {
```

```
  string st;
  vector<ll> dir, rev;
  vector<ll> pws;
  ll n;
  ll mod = 0, base = 0;

  StringHash(string st, ll base, ll mod): st(st), n(st.size()), base(base
    ), mod(mod) {
    dir.resize(n + 1);
    rev.resize(n + 1);
    pws.resize(n + 1, 1);
    for (int i = 1; i < n + 1; i++) {
      pws[i] = pws[i - 1] * base;
      pws[i] %= mod;
    }

    for (int i = 0; i < n; i++) {
      dir[i + 1] = dir[i] + (st[i] - '0' + 1) * pws[i];
      rev[i + 1] = rev[i] * base + (st[i] - '0' + 1);
      dir[i + 1] %= mod;
      rev[i + 1] %= mod;
    }
  }

  ll getHash() {
    return rev[n];
  }

  ll getCycle(ll beg) {
    ll fir = (rev[n] - ((rev[beg] * pws[n - beg]) % mod) + mod) % mod;
    ll sec = rev[beg];
    return (((fir * pws[beg]) % mod) + sec) % mod;
  }
};

int main()
{
  cin.tie(0);
  ios_base::sync_with_stdio(false);
  string type; cin >> type;
  if (type == "first") {
    ll n; cin >> n;
    string st; cin >> st;
    StringHash ha1(st, BASE1, MOD1);
    //StringHash ha2(st, BASE2, MOD2);
    //StringHash ha3(st, BASE3, MOD3);
    //cout << ha1.getHash() << " " << ha2.getHash() << " " << ha3.getHash
    ();
    cout << ha1.getHash();
  }
  else {
    ll n; cin >> n;
    string st; cin >> st;
```

```
    ll h1, h2, h3; cin >> h1 >> h2 >> h3;
    if (h1 < 0 || h2 < 0 || h3 < 0) return -192;
    StringHash ha1(st, BASE1, MOD1);
    //StringHash ha2(st, BASE2, MOD2);
    //StringHash ha3(st, BASE3, MOD3);
    for (int i = 0; i < n; i++) {
      ll k1 = ha1.getCycle(i);
      //ll k2 = ha2.getCycle(i);
      //ll k3 = ha3.getCycle(i);
      //if (k1 == h1 && k2 == h2 && k3 == h3) {
      if (k1 == h1) {
        cout << "YES" << endl;
        return 0;
      }
    }
    cout << "NO" << endl;
  }
}
```

Task D

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <set>

using namespace std;
typedef long long ll;

vector<ll> state;

bool wait() {
  vector<ll> state2;
  for (int i = 1; i < state.size(); i++) state2.push_back(state[i]);
  state2.push_back(state[0]);
  state = state2;

  cout << "1 1" << endl;
  string st; cin >> st;
  if (st == "!") exit(0);
  return st == "<";
}

void change(ll i1, ll i2) {
  ll tmp = state[i1];
  state[i1] = state[i2];
  state[i2] = tmp;
  vector<ll> state2;
  for (int i = 1; i < state.size(); i++) state2.push_back(state[i]);
  state2.push_back(state[0]);
  state = state2;
```

```
  cout << (i1 + 1) << " " << (i2 + 1) << endl;
  string st; cin >> st;
  if (st == "!") exit(0);

}

ll getPos(ll i) {
  return state[i];
}

vector<set<ll>> gr;
vector<set<ll>> gr2;

void pushEdge(ll v, ll u) {
  if (v == u) return;
  gr[v].insert(u);
  gr2[u].insert(v);
  //cout << u + 1 << " is less than " << v + 1 << endl;
}

vector<bool> used1, used2;
ll glmin = -1;
void dfs1(ll v) {
  used1[v] = true;
  if (gr[v].size() == 0) glmin = v;
  for (ll to : gr[v]) {
    if (!used1[to]) dfs1(to);
  }
}
int main()
{
  //cin.tie(0);
  //ios_base::sync_with_stdio(false);
  ll n; cin >> n;
  string dd; cin >> dd;
  state.resize(n, 0);
  gr.resize(n);
  gr2.resize(n);
  used1.resize(n);
  used2.resize(n);
  for (int i = 0; i < n; i++) state[i] = i;
  ll curnum = 0;
  for (ll el = 0; el < n; el++) {
    for (int i = 0; i < n; i++) {
      bool less = wait();
      if (less) {
        pushEdge(curnum, getPos(0));
      }
      else {
        pushEdge(getPos(0), curnum);
      }
      if (n % 2 == 0 && i == n / 2 - 1) {
        change(n - 1, 0);
```

```
      change(n - 1, 1);
    }
    else {
      if (i != n - 1) {
        change(n - 1, 1);
      }
    }
  }
  for (int i = 0; i < n; i++) {
    if (state[i] == curnum + 1) {
      curnum++;
      change(i, 1);
    }
  }
}
dfs1(0);
set<pair<ll, ll>> qu;
qu.insert({ 1, glmin });
vector<pair<ll, ll>> dist(n, {-1, -1});
dist[glmin] = { 1, -1 };
while (!qu.empty()) {
  auto[_, v] = *qu.begin();
  qu.erase(qu.begin());
  for (ll to : gr2[v]) {
    if (dist[v].first + 1 > dist[to].first) {
      dist[to].first = dist[v].first + 1;
      dist[to].second = v;
      qu.insert({ dist[to].first, to });
    }
  }
}
ll ma = 0, maid = 0;
for (int i = 0; i < n; i++) {
  if (dist[i].first > ma) {
    ma = dist[i].first;
    maid = i;
  }
}
ll curr = maid;
vector<ll> path;
while (curr != -1) {
  path.push_back(curr);
  curr = dist[curr].second;
}

for (int i = 0; i < n; i++) {

  ll val = path[i];
  ll ind = -1;
  for (int j = 0; j < n; j++) {
    if (state[j] == val) {
      ind = j;
      break;
```

```
      }
    }
    change(ind, 0);
  }
  //dd
  while (true) {
    wait();
  }
  return 0;
}
```

Task E

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>

using namespace std;
typedef long long ll;

vector<ll> ord1, ord2;
vector<vector<ll>> gr;
vector<ll> pr;
vector<bool> used;
vector<ll> pnts;
ll mrk = 0;

void dfs1(ll v, ll p) {
  if (!used[v]) mrk++;
  used[v] = true;
  ord1.push_back(v);
  ll i = pnts[v];
  while (true) {
    ll to = gr[v][i];
    if (to == p) {
      pr[v] = p;
      pnts[v] = (i + 1) % gr[v].size();
      return;
    }
    dfs1(to, v);
    ord1.push_back(v);
    i++;
    i %= gr[v].size();
    if (v == 0 && (ord1.size() > 1e6 + 100 || (mrk == gr.size() && i ==
   0))) {
      ord1.pop_back();
      return;
    }
  }
  int dbg = 0;
}
```

```
void dfs2(ll v, ll p) {
  ord2.push_back(v);
  for (ll to : gr[v]) {
    if (to == p) return;
    dfs2(to, v);
    ord2.push_back(v);
  }
}

int main()
{
  cin.tie(0);
  ios_base::sync_with_stdio(false);
  ll n, q; cin >> n >> q;
  gr.resize(n);
  pr.resize(n);
  used.resize(n);
  pnts.resize(n, 0);
  for (int i = 0; i < n; i++) {
    ll k; cin >> k;
    for (int j = 0; j < k; j++) {
      ll w; cin >> w;
      gr[i].push_back(w - 1);
    }
  }
  dfs1(0, -1);
  if (mrk == n) {
    for (int i = 1; i < n; i++) {
      ll ind = -1;
      for (int j = 0; j < gr[i].size(); j++) {
        if (gr[i][j] == pr[i]) {
          ind = j;
          break;
        }
      }
      vector<ll> nw;
      for (int j = ind + 1; j < gr[i].size(); j++) {
        nw.push_back(gr[i][j]);
      }
      for (int j = 0; j < ind; j++) {
        nw.push_back(gr[i][j]);
      }
      nw.push_back(gr[i][ind]);
      gr[i] = nw;
    }
    dfs2(0, -1);
    ord2.pop_back();
  }
  while (q--) {
    ll x; cin >> x;
    if (x < ord1.size()) {
      cout << ord1[x]+1 << " ";
    }
```

```
    else {
      x -= ord1.size();
      x %= ord2.size();
      cout << (ord2[x]+1) << " ";
    }
  }
  cout << endl;
  //diuiuwe
}
```

Task F

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <set>

using namespace std;
typedef long long ll;

int main()
{
  cin.tie(0);
  ios_base::sync_with_stdio(false);
  ll w, h; cin >> w >> h;
  vector<vector<ll>> ar(w, vector<ll>(h));
  for (int i = 0; i < w; i++) {
    string st; cin >> st;
    for (int j = 0; j < h; j++) {
      ar[i][j] = st[j] - '0';
    }
  }
  ll ans = 0;
  for (int i = 0; i < w; i++) {
    for (int j = 0; j < h/2; j++) {
      ans += ar[i][j * 2] * ar[i][j * 2 + 1];
    }
  }
  cout << ans << endl;
  for (int i = 0; i < w; i++) {
    for (int j = 0; j < h/2; j++) {
      cout << "LR";
    }
    cout << endl;
  }
}
```