Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0137 – Дидык Иван Сергеевич

A	B	C	D	E	F	sum
100	100	100	100	40	-	440

Task A

```
#include <bits/stdc++.h>
using namespace std;
#define vec vector
#define umap unordered_map
#define uset unordered_set
#define ll long long
#define all(v) v.begin(), v.end()
#define forn(i, n) for(int i = 0; i < n; ++i)

void solve() {
  int a, b, n; cin >> a >> b >> n;

  if (a > n) {
    cout << "0" << endl;
    return;
  } else if (b > n) {
    if (n % a == 0) cout << a << endl;
    else cout << "0" << endl;
    return;
  }

  ll res = 0;
  forn(st, b) {
    vec<ll> dp(n - st + 1, 0);
    dp[0] = 1;
    for(int i = 1; i < dp.size(); ++i) {
      if (i - a >= 0) dp[i] += dp[i-a];
      if (i - b >= 0) dp[i] += dp[i-b];
    }
```

```
    if (st < a)
      res += dp[n - a];
    res += dp[n - b];

  }
  cout << res << endl;
}

int main() {
  ios::sync_with_stdio(false);
  cin.tie(0);
  solve();
  return 0;
}
```

Task B

```
#include <bits/stdc++.h>
using namespace std;
#define vec vector
#define umap unordered_map
#define uset unordered_set
#define ll long long
#define all(v) v.begin(), v.end()
#define forn(i, n) for(int i = 0; i < n; ++i)

struct v3 {
  int x, y, z;
};

int dist(v3 a, v3 b) {
  return (abs(a.x - b.x) + abs(a.y - b.y) + abs(a.z - b.z)) / 2;
}

void solve() {
  int n; cin >> n;
  v3 a, b;
  cin >> a.x >> a.y >> a.z;
  cin >> b.x >> b.y >> b.z;
  int md = dist(a, b);
  md = min(md, dist(a, v3 { b.x - n*2, b.y + n, b.z + n }));
  md = min(md, dist(a, v3 { b.x + n, b.y - n*2, b.z + n }));
  md = min(md, dist(a, v3 { b.x + n, b.y + n, b.z - n*2 }));
  md = min(md, dist(a, v3 { b.x + n*2, b.y - n, b.z - n }));
  md = min(md, dist(a, v3 { b.x - n, b.y + n*2, b.z - n }));
  md = min(md, dist(a, v3 { b.x - n, b.y - n, b.z + n*2 }));
  cout << md << endl;

}

int main() {
  ios::sync_with_stdio(false);
  cin.tie(0);
```

```
  solve();
  return 0;
}
```

Task C

```
#include <bits/stdc++.h>
using namespace std;
#define vec vector
#define umap unordered_map
#define uset unordered_set
#define ll long long
#define all(v) v.begin(), v.end()
#define forn(i, n) for(int i = 0; i < n; ++i)

template<ll mod>
struct phash {
  vec<ll> pows, val;
  phash(const string &s) {
    val.resize(s.size()+1);
    pows.resize(s.size()+1);
    pows[0] = 1;
    val[0] = 0;
    forn(i, s.size()) {
      val[i+1] = (val[i] * 3 + (s[i] - '0') + 1) % mod;
      pows[i+1] = pows[i] * 3 % mod;
    }
  }
  ll operator()(int l, int r) {
    return (val[r] + mod - val[l] * pows[r-l] % mod) % mod;
  }
};

const ll m1 = 1e9+7,
         m2 = 1e9+9;

void solve() {
  string t; cin >> t;
  int n; cin >> n;
  string s; cin >> s;
  n = s.size();
  phash<m1> h1(s);
  phash<m2> h2(s);

  if (t == "first") {
    cout << h1(0, s.size()) << ' ' << h2(0, s.size()) << endl;
  } else {
    ll hv1, hv2;
    cin >> hv1 >> hv2;
    forn(i, n) {
      ll hc1 = (h1(i, n) * h1.pows[i] + h1(0, i)) % m1;
      ll hc2 = (h2(i, n) * h2.pows[i] + h2(0, i)) % m2;
      //cout << "off " << i << " h " << hc1 << " " << hc2 << endl;
```

```
      if (hv1 == hc1 && hv2 == hc2) {
        cout << "YES\n";
        return;
      }
    }
    cout << "NO\n";
  }
}

int main() {
  ios::sync_with_stdio(false);
  cin.tie(0);
  solve();
  return 0;
}
```

Task D

```
#include <bits/stdc++.h>
using namespace std;
#define vec vector
#define umap unordered_map
#define uset unordered_set
#define ll long long
#define all(v) v.begin(), v.end()
#define forn(i, n) for(int i = 0; i < n; ++i)
#ifdef DEBUG
  #define debug if(1)
#else
  #define debug if(0)
#endif

// <-
int shift = 0, n;
vec<int> ids;
// id -> index in ids
vec<int> id_at;

// seq pos
void sswap(int a, int b) {
  id_at[ids[a]] = b;
  id_at[ids[b]] = a;
  swap(ids[a], ids[b]);
}


void _log() {
  debug {
    cerr << "== Perm: ";
    forn(i, n) cerr << ids[(i + shift) % n] << ' ';
    cerr << endl;
  }
}
```

```
// [id] < [id]
bool is_less(int a, int b) {
  int pa = (id_at[a] - shift + 2*n) % n;
  cout << "2 " << pa+1 << endl;
  sswap(id_at[a], (shift + 1) % n);
  shift = (shift+1)%n;
  char res; cin >> res;
  if (res == '!') exit(0);
  _log();

  int pb = (id_at[b] - shift + 2*n) % n;
  cout << "2 " << pb+1 << endl;
  sswap(id_at[b], (shift + 1) % n);
  shift = (shift+1)%n;
  cin >> res;
  if (res == '!') exit(0);
  _log();
  return res == '>';
}

// pos in id
void seq_swap(int a, int b) {
  cout << (2*n - shift + a) % n + 1 << " " << (2*n - shift + b) % n + 1
    << endl;
  char res; cin >> res;
  sswap(a, b);
  shift = (shift+1)%n;
  if (res == '!') exit(0);
  _log();
}

void solve() {
  cin >> n;
  ids.resize(n); id_at.resize(n);
  iota(all(ids), 0);
  iota(all(id_at), 0);

  vec<int> cnt_gt(n);

  forn(i, n) {
    for(int j = i+1; j < n; ++j) {
      //cerr << "=== ? " << i << " " << j << endl;
      bool v = is_less(i, j);
      if (v) ++cnt_gt[i];
      else ++cnt_gt[j];
      //cerr << "=== seq " << i << " < " << j << ": " << v << endl;
    }
  }
  debug {
    cerr << "=== cnt less: ";
    for(auto i : cnt_gt) cerr << i << ' ';
    cerr << endl;
```

```
    cerr << "== ids: ";
    for(auto i : ids) cerr << i << ' ';
    cerr << endl;
  }

  forn(i, n) {
    if (id_at[i] != cnt_gt[i]) {
      debug cerr << "=== " << i << " at " << id_at[i] << " instead of "
    << cnt_gt[i] << endl;
      seq_swap(cnt_gt[i], id_at[i]);
    }
  }

  forn(i, n - shift) {
    cout << "1 1" << endl;
    char c; cin >> c;
    if (c == '!') exit(0);
  }

  cout << "end" << endl;
}

int main() {
  ios::sync_with_stdio(false);
  cin.tie(0);
  solve();
  return 0;
}
```

Task E

```
#include <bits/stdc++.h>
using namespace std;
#define vec vector
#define umap unordered_map
#define uset unordered_set
#define ll long long
#define all(v) v.begin(), v.end()
#define forn(i, n) for(int i = 0; i < n; ++i)
#ifdef DEBUG
  #define debug if(1)
#else
  #define debug if(0)
#endif


struct qs {
  ll t;
  int idx;
  bool operator<(qs o) { return t < o.t; }
};

int n;
```

```
vec<vec<int>> G;
vec<vec<int>> G_ord;
vec<int> path;

void dfs_path(int v, int p) {
  path.push_back(v);
  int sz = G[v].size();
  int idx = v == p ? 0 : (find(all(G[v]), p) - G[v].begin() + 1) % sz;
  for(int _cnt = 0; _cnt < (v == p ? sz : sz-1); _cnt++, idx = (idx+1) %
    sz) {
    dfs_path(G[v][idx], v);
    path.push_back(v);
    G_ord[v].push_back(G[v][idx]);
  }
}

vec<bool> visits;
vec<qs> qss;
vec<int> ans;
int qp;
ll curr_time;
int fin_cnt = 0;

void dfs_calc(int v, int p, int pd=0) {
  while (qss[qp].t == curr_time) {
    ans[qss[qp].idx] = v;
    ++qp;
  }
  ++curr_time;
  for (auto i : (visits[v] ? G_ord[v] : G[v])) {
    if (i == p) break;
    dfs_calc(i, v, pd+1);
    while (qss[qp].t == curr_time) {
      ans[qss[qp].idx] = v;
      ++qp;
    }
    ++curr_time;
  }
  if (!visits[v]) ++fin_cnt;
  visits[v] = true;
}

void solve() {
  int q; cin >> n >> q;
  G.resize(n);
  G_ord.resize(n);
  visits.assign(n, false);
  qss.resize(q);
  ans.assign(q, -1);

  forn(i, n) {
    int c; cin >> c;
    G[i].resize(c);
```

```
    forn(j, c) {
      cin >> G[i][j];
      --G[i][j];
    }
  }

  forn(i, q) {
    cin >> qss[i].t;
    qss[i].idx = i;
  }
  sort(all(qss));
  qss.push_back({ .t=LLONG_MAX, .idx = -1 });
  qp = 0;
  curr_time = 0;

  dfs_path(0,0);

  while (qp < q) {
    dfs_calc(0, 0);
    --curr_time;
    if (fin_cnt == n) break;
  }
  ++curr_time;
  for (;qp < q; ++qp)
    ans[qss[qp].idx] = path[1 + (qss[qp].t - curr_time) % (path.size() -
   1)];

  for (auto i : ans) {
    cout << i+1 << ' ';
  }
  cout << endl;
}

int main() {
  ios::sync_with_stdio(false);
  cin.tie(0);
  solve();
  return 0;
}
```