Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0172 – Зедгенизова Маргарита Дмитриевна

A	B	C	D	E	F	sum
100	100	100	100	20	51	471

Task A

```
#include <iostream>
#include <vector>
using namespace std;
using ll = long long;

int main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  int a, b, n;
  cin >> a >> b >> n;
  vector<ll> dp(n + 1, 0);
  dp[0] = 1;
  for (int i = 1; i <= n; ++i) {
    if (i - a >= 0)
      dp[i] += dp[i - a];
    if (i - b >= 0)
      dp[i] += dp[i - b];
  }
  ll ans = 0;
  if (n - a >= 0)
    ans += dp[n - a] * a;
  if (n - b >= 0)
    ans += dp[n - b] * b;
  cout << ans;

  return 0;
}
```

Task B

```
#include <iostream>
#include <vector>
#include <map>
using namespace std;
using ll = long long;

int get_dist(int x1, int y1, int x2, int y2) {
  int dx = x1 - x2;
  int dy = y1 - y2;
  int ans;
  if (dx <= 0 && dy >= 0) {
    ans = max(abs(dx), dy);
  }
  else if (dx <= 0 && dy <= 0) {
    ans = abs(dx) + abs(dy);
  }
  else if (dx >= 0 && dy >= 0) {
    ans = dx + dy;
  }
  else {
    ans = max(dx, abs(dy));
  }
  return ans;
}

int main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  int n;
  cin >> n;
  int x1, y1, z1, x2, y2, z2;
  cin >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
  int ans = get_dist(x1, y1, x2, y2);
  ans = min(ans, get_dist(x1, y1, x2 + (2 * n), y2 - n));
  ans = min(ans, get_dist(x1, y1, x2 - (2 * n), y2 + n));
  ans = min(ans, get_dist(x1, y1, x2 - n, y2 + (2 * n)));
  ans = min(ans, get_dist(x1, y1, x2 + n, y2 - (2 * n)));
  ans = min(ans, get_dist(x1, y1, x2 + n, y2 + n));
  ans = min(ans, get_dist(x1, y1, x2 - n, y2 - n));
  cout << ans;

  return 0;
}
```

Task C

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

```
const int sz = 10000;
const ll MOD = 1e9 + 7, BASE = 31;

int get_hash(const string& s) {
  ll ans = 0;
  for (int i = 0; i < s.size(); ++i) {
    ans = (ans * BASE + (s[i] - '0' + 1)) % MOD;
  }
  return ans;
}

int main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  string type;
  cin >> type;
  if (type == "first") {
    int n;
    cin >> n;
    string s;
    cin >> s;
    string cur = s;
    for (int i = 0; i < n; ++i) {
      cur = cur.substr(1, n - 1) + cur[0];
      s = max(s, cur);
    }
    string ans = "";
    for (int i = 0; i < n; i += sz) {
      ans += to_string(get_hash(s.substr(i, sz)));
    }
    cout << ans;
  }
  else {
    int n;
    cin >> n;
    string s, ans1;
    cin >> s >> ans1;
    string cur = s;
    for (int i = 0; i < n; ++i) {
      cur = cur.substr(1, n - 1) + cur[0];
      s = max(s, cur);
    }
    string ans = "";
    for (int i = 0; i < n; i += sz) {
      ans += to_string(get_hash(s.substr(i, sz)));
    }
    if (ans == ans1)
      cout << "YES";
    else
      cout << "NO";
  }
```

```
  return 0;
}
```

Task D

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <random>
using namespace std;
using ll = long long;

int n;
int gnext2(int i, int c) {
  if (i - c >= 1)
    return i - c;
  c %= n;
  return i + n - c;
}

int main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  cin >> n;
  vector<int> ind(n + 1);
  vector<int> mas(n + 1);
  for (int i = 1; i <= n; ++i) {
    ind[i] = i;
    mas[i] = i;
  }
  vector<int> val(n + 1, 1);
  for (int cur = 1; cur <= n; ++cur) {
    for (int cur2 = 1; cur2 <= n; ++cur2) {
      if (cur2 == cur)
        continue;
      int i = ind[cur], j = 2;
      cout << i << " " << j << endl;
      swap(mas[i], mas[j]);
      ////////////////////////////////////////////////////
      int pred = mas[1];
      for (int k = 1; k < n; ++k) {
        mas[k] = mas[k + 1];
        ind[mas[k]] = k;
      }
      mas[n] = pred;
      ind[mas[n]] = n;
      char ans; cin >> ans;
      if (ans == '!')
        return 0;
      ////////////////////////////////////////////////////
      i = ind[cur2], j = 2;
```

```
      cout << i << " " << j << endl;
      swap(mas[i], mas[j]);
      ////////////////////////////////////////////////////
      pred = mas[1];
      for (int k = 1; k < n; ++k) {
        mas[k] = mas[k + 1];
        ind[mas[k]] = k;
      }
      mas[n] = pred;
      ind[mas[n]] = n;
      cin >> ans;
      if (ans == '<')
        val[cur]++;
      if (ans == '!')
        return 0;
    }
  }
  vector<int> ind2(n + 1);
  for (int i = 1; i <= n; ++i) {
    ind2[val[i]] = i;
  }
  //vector<int> m = { 0, 2, 5, 3, 4, 1 };
  ///for (int i = 1; i <= n; ++i) {
    //ind2[m[i]] = i;
  //}
  int cur = 0;
  for (int v = n; v >= 1; --v) {
    int i = ind[ind2[v]], j = gnext2((n - v + 1), cur);
    cout << i << " " << j << endl;
    swap(mas[i], mas[j]);
    int pred = mas[1];
    for (int k = 1; k < n; ++k) {
      mas[k] = mas[k + 1];
      ind[mas[k]] = k;
    }
    mas[n] = pred;
    ind[mas[n]] = n;
    char ans;
    cin >> ans;
    if (ans == '!')
      return 0;
    cur++;
  }
  while (true) {
    cout << "1 1" << endl;
    char ans; cin >> ans;
    if (ans == '!')
      return 0;
  }

  return 0;
}
```

Task E

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <iomanip>
#include <random>
using namespace std;
using ll = long long;
const int sz = 1e6 + 1;
const ll MOD = 1e9 + 7, BASE = 31;

int main() {
  ios_base::sync_with_stdio(0);
  cin.tie(0);

  int n, q;
  cin >> n >> q;
  vector<vector<int>> g(n);
  for (int i = 0; i < n; ++i) {
    int cnt; cin >> cnt;
    g[i].resize(cnt);
    for (int j = 0; j < cnt; ++j) {
      cin >> g[i][j];
      g[i][j]--;
    }
  }
  vector<int> sdv(n, 0);
  vector<int> ans(sz);
  ans[0] = 1;
  int u = 0;
  for (int t = 1; t < sz; ++t) {
    int v = g[u][sdv[u] % (g[u].size())];
    sdv[u]++;
    u = v;
    ans[t] = u + 1;
  }
  vector<ll> h(sz);
  vector<ll> base(sz, 1);
  h[0] = 1;
  for (int i = 1; i < sz; ++i) {
    h[i] = (h[i - 1] * BASE + ans[i]) % MOD;
    base[i] = base[i - 1] * BASE % MOD;
  }
  bool flag = false;
  int start = sz + 1, siz;
  for (int j = 1; j < 100; ++j) {
    for (int i = 1000; 2 * i + j - 1 < sz; ++i) {
      ll h1 = (h[i + j - 1] - h[j - 1] * base[j] % MOD + MOD) % MOD;
      ll h2 = (h[2 * i + j - 1] - (h[i + j - 1] * base[i + j]) % MOD +
   MOD) % MOD;
      if (h1 == h2) {
        flag = true;
```

```
        start = j;
        siz = i;
        break;
      }
    }
    if (flag)
      break;
  }
  for (int _ = 0; _ < q; ++_) {
    int t;
    cin >> t;
    int res;
    if (t < start) {
      res = ans[t];
    }
    else {
      t -= start;
      res = ans[start + t % siz];
    }
    cout << res << " ";
  }

  return 0;
}
```

Task F

```
for (int i = 0; i < n; i += 2) {
    for (int j = 0; j < m; j += 2) {
      int add1 = mas[i][j] * mas[i][j + 1] + mas[i + 1][j] * mas[i + 1][j
     + 1];
      int add2 = mas[i][j] * mas[i + 1][j] + mas[i][j + 1] * mas[i + 1][j
     + 1];
      if (add1 >= add2) {
        ans += add1;
        ans_c[i][j] = 'L';
        ans_c[i][j + 1] = 'R';
        ans_c[i + 1][j] = 'L';
        ans_c[i + 1][j + 1] = 'R';
      }
      else {
        ans += add2;
        ans_c[i][j] = 'U';
        ans_c[i][j + 1] = 'U';
        ans_c[i + 1][j] = 'B';
        ans_c[i + 1][j + 1] = 'B';
      }
    }
  }
```