Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0273 – Лычагин Илья Витальевич

A	B	C	D	E	F	sum
100	100	100	100	20	57	477

Task A

```
def f(a):
  cur = 1
  for i in range(1, a + 1):
    cur = cur * i
  return cur


Q = input()
Q = Q.split(' ')
a = int(Q[0])
b = int(Q[1])
n = int(Q[2])
plus1 = 0
plus2 = 0
for i in range(n + 1):
  c = i
  if (i * a > n):
    break
  if ((n - i * a) % b == 0):
    d = (n - i * a) // b;
    plus1 = plus1 + f(d + c) // f(d) // f(c);

n = n - b;
for i in range(n + 1):
  c = i
  if (i * a > n):
    break
  if ((n - i * a) % b == 0):
```

```
    d = (n - i * a) // b;
    plus2 = plus2 + f(d + c) // f(d) // f(c);
z = 0;
ans = 0;
for i in range(b):
  if (z < a):
    ans = ans + plus1;
  else:
    ans = ans + plus2;
  z += 1
print(ans)
```

Task B

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <math.h>
#include <iomanip>
#include <set>

using namespace std;

long long x1, yl, z1;

long long find(long long x2, long long y2, long long z2) {
  return min(abs(x1 - x2) + abs(yl - y2), min(abs(yl - y2) + abs(z1 - z2)
    , abs(z1 - z2) + abs(x1 - x2)));
}

int main() {
  ios::sync_with_stdio(0);
  cin.tie(0);
  long long n;
  long long x2, y2, z2;
  cin >> n;
  cin >> x1 >> yl >> z1;
  cin >> x2 >> y2 >> z2;
  long long c1 = find(x2, y2, z2);
  c1 = min(c1, find(x2 - n, y2 - n, z2 + 2 * n));
  c1 = min(c1, find(x2 - 2 * n, y2 + n, z2 + n));
  c1 = min(c1, find(x2 + n, y2 - 2 * n, z2 + n));
  c1 = min(c1, find(x2 + 2 * n, y2 - n, z2 - n));
  c1 = min(c1, find(x2 - n, y2 + 2 * n, z2 - n));
  c1 = min(c1, find(x2 + n, y2 + n, z2 - 2 * n));
  cout << c1;
}
```

Task C

```
#include <iostream>
#include <vector>
```

```
#include <algorithm>
#include <math.h>
#include <iomanip>
#include <queue>

using namespace std;

long long Mod = 1e9 + 7;
long long p = 121;

long long sum(long long a, long long b) {
  if (a + b >= Mod) {
    return a + b - Mod;
  }
  return a + b;
}

long long sub(long long a, long long b) {
  if (a - b < 0) {
    return a - b + Mod;
  }
  return a - b;
}

int main() {
  ios::sync_with_stdio(0);
  cin.tie(0);
  string t;
  cin >> t;
  vector <long long> pows(100001);
  pows[0] = 1;
  for (int i = 1; i <= 100000; ++i) {
    pows[i] = pows[i - 1] * p % Mod;
    //cout << pows[i] << " ";
  }
  if (t == "first") {
    int n;
    cin >> n;
    string s;
    cin >> s;
    long long Hash = 0;
    for (int i = 0; i < n; ++i) {
      Hash = sum(Hash * p % Mod, s[i] - '0' + 1);
    }
    cout << Hash;
  } else {
    int n;
    cin >> n;
    string t;
    cin >> t;
    long long Hash;
    cin >> Hash;
    long long H = 0;
```

```
    for (int i = 0; i < n; ++i) {
      H = sum(H * p % Mod, t[i] - '0' + 1);
    }
    if (H == Hash) {
      cout << "YES";
      return 0;
    }
    for (int i = 1; i < n; ++i) {
      H = sub(H, (t[i - 1] - '0' + 1) * pows[n - 1] % Mod);
      H = H * p % Mod;
      H = sum(H, t[i - 1] - '0' + 1);
      if (H == Hash) {
        cout << "YES";
        return 0;
      }
      //cout << H << " H " << endl;
    }
    cout << "NO";
    return 0;
  }
}
```

Task D

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <fstream>
#include <queue>
#include <iomanip>
#include <random>
#include <time.h>
#include <map>

using namespace std;

int main() {
  int n;
  cin >> n;
  //string S;
  //cin >> S;
  vector <pair <int, int>> move(n * n + 3, {1, 1});
  for (int i = 0; i <= n * n - 1; ++i) {
    if (i % n == 0) {
      cout << 1 << " " << 1 << endl;
    } else {
      cout << move[i].first << " " << move[i].second << endl;
      //swap(S[move[i].first - 1], S[move[i].second - 1]);
    }
    //string S1 = S;
    //S = string(S.begin() + 1, S.end()) + S[0];
    //cout << S  << endl;
    char t;
```

```
    cin >> t;
    /*if (S > S1) {
      t = '>';
    } else {
      t = '<';
    }*/
    if (t == '>') {
      move[i + 1] = {1, n};
    }
  }
  cout << "!" << endl;
  //cout << S;
}
```

Task E

```
#include <iostream>
#include <algorithm>
#include <vector>
#include <fstream>
#include <queue>
#include <iomanip>
#include <random>
#include <time.h>
#include <map>

using namespace std;

int n, q;
vector <vector <int>> g;

vector <int> T;
vector <int> p;
vector <int> used;
int t = 0;
bool flag = true;

int NoBlock;

int St = 0;

void dfs(int cur, int Par) {
  //cout << cur + 1 << " cur " << NoBlock << endl;
  //cout << cur + 1 << " ";
  T[t] = cur;
  t += 1;
  while (true) {
    if (NoBlock <= 0 && cur == 0 && p[cur] == 0) {
      //cout << " h " << endl;
      St = t;
      return;
    }
    if (g[cur][p[cur]] == Par && used[cur] == 0) {
```

```
      used[cur] = 1;
      p[cur] = (p[cur] + 1) % g[cur].size();
      NoBlock -= 1;
      if (NoBlock == 0) {
        St = t;
      }
      return;
    }
    if (g[cur][p[cur]] == Par) {
      p[cur] = (p[cur] + 1) % g[cur].size();
      return;
    }
    int next = g[cur][p[cur]];
    p[cur] = (p[cur] + 1) % g[cur].size();
    dfs(next, cur);
    //cout << cur << " cur " << endl;
    //cout << cur + 1 << " ";
    T[t] = cur;
    t += 1;
  }
}

vector <int> T2;

void dfs2(int cur, int par) {
  //cout << cur << " cur " << par << endl;
  T2[t] = cur;
  t += 1;
  for (auto ti : g[cur]) {
    if (ti == par) {
      continue;
    }
    dfs2(ti, cur);
    T2[t] = cur;
    t += 1;
  }
}

int main() {
  cin >> n >> q;
  NoBlock = n - 1;
  used.resize(n, 0);
  g.resize(n);
  p.resize(n, 0);
  T.resize(1000001, 0);
  T2.resize(1000001, 0);
  for (int i = 0; i < n; ++i) {
    int sz;
    cin >> sz;
    for (int j = 0; j < sz; ++j) {
      int v;
      cin >> v;
      --v;
```

```
      g[i].push_back(v);
    }
  }
  dfs(0, -1);
  t = 0;
  dfs2(0, -1);
  //cout << St << " S " << t << " t " << endl;
  //cout << t << " t " << St << " St " << endl;
  for (int i = 0; i < q; ++i) {
    long long Cur;
    cin >> Cur;
    if (Cur < St) {
      cout << T[Cur] + 1 << " ";
    } else {
      cout << T2[(Cur - St + 1) % t] + 1 << " ";
    }

  }
}
```

Task F

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <random>

using namespace std;

mt19937 rnd(1);

int sub(int a, int b, int mod) {
    if (a - b < 0) {
        return a - b + mod;
    }
    return a - b;
}

int sum(int a, int b, int mod) {
    if (a + b >= mod) {
        return a + b - mod;
    }
    return a + b;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    int h, w;
    cin >> h >> w;
    vector <vector <long long>> A(h, vector <long long> (w));
    for (int i = 0; i < h; ++i) {
```

```
    for (int j = 0; j < w; ++j) {
        char t;
        cin >> t;
        A[i][j] = t - '0';
    }
}
vector <vector <char>> BestT;
long long BestAns = 0;
vector <vector <char>> T(h, vector <char> (w, '.'));
for (int k = 0; k < 50000; ++k) {
T = {};
T.resize(h);
for (int j = 0; j < h; ++j) {
    T[j].resize(w, '.');
}
long long ans = 0;
for (int i = 0; i < h; ++i) {
    for (int j = 0; j < w; ++j) {
        if (T[i][j] != '.') {
            continue;
        }
        int d = rnd() % 4;
        if (d == 0 && T[i][sub(j, 1, w)] == '.') {
            T[i][j] = 'R';
            T[i][sub(j, 1, w)] = 'L';
            ans = ans + A[i][j] * A[i][sub(j, 1, w)];
        } else if (d == 0) {
            d = (d + 1) % 4;
        }

        if (d == 1 && T[sub(i, 1, h)][w] == '.') {
            T[i][j] = 'B';
            T[sub(i, 1, h)][w] = 'T';
            ans = ans + A[i][j] * A[sub(i, 1, h)][w];
        } else if (d == 1) {
            d = (d + 1) % 4;
        }

        if (d == 2 && T[i][sum(j, 1, w)] == '.') {
            T[i][j] = 'L';
            T[i][sum(j, 1, w)] = 'R';
            ans = ans + A[i][j] * A[i][sum(j, 1, w)];
        } else if (d == 2) {
            d = (d + 1) % 4;
        }

        if (d == 3 && T[sum(i, 1, h)][j] == '.') {
            T[i][j] = 'T';
            T[sum(i, 1, h)][j] = 'B';
            ans = ans + A[i][j] * A[sum(i, 1, h)][j];
        } else if (d == 3) {
            d = (d + 1) % 4;
        }
```

```
        }
    }
    if (ans > BestAns) {
        BestAns = ans;
        BestT = T;
    }
    }
    cout << BestAns << endl;
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            cout << BestT[i][j];
        }
        cout << endl;
    }
}
```