Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0408 – Селезнев Никита Александрович

A	B	C	D	E	F	sum
100	100	100	-	20	67	387

Task A

```
#include <iostream>

using namespace std;
using ll = long long int;

const int maxn = 1800;

ll dp[maxn] = {0};

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

Task B

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

using namespace std;
using ll = long long int;

const int maxn = 1800;

struct Num{
    ll x, y, z;
};
```

```
int main() {
    ll n, x1, y1, z1, x2, y2, z2;
    cin >> n >> x1 >> y1 >> z1 >> x2 >> y2 >> z2;
    vector <Num> a = {
        {x1, y1, z1},
        {x1+n*2, y1-n, z1-n},
        {x1-n, y1+2*n, z1-n},
        {x1-n, y1-n, z1+2*n},
        {x1-n*2, y1+n, z1+n},
        {x1+n, y1-2*n, z1+n},
        {x1+n, y1+n, z1-2*n},
    };
    ll ans = -1;
    for (auto i : a) {
        ll res = ll((abs(i.x-x2) + abs(i.y-y2) + abs(i.z-z2))/2);
        ans = (ans > res || ans == -1 ? res : ans);
    }
    cout << ans;
}
```

Task C

```
MOD = 10 ** 9 + 7
p = 2017

kpow = [1]
for _ in range(100000*2):
    kpow.append((kpow[-1]*p) % MOD)

c = input()
n = int(input())
s = input()
if (c == 'first'):
    hash = 0
    for i, num in enumerate(s):
        hash = (hash + int(num) * kpow[i]) % MOD
    print(hash)
else:
    hash = int(input())
    currHash = 0
    s = s + s
    #print(s)
    for i, num in enumerate(s):
        if (i >= n):
            if (s[i-n]=='1'): currHash = (currHash + MOD - (kpow[(i - n)]
     % MOD)) % MOD
            #print("!", currHash)
        currHash = (currHash + int(num) * kpow[i]) % MOD
        #print(i, currHash, (hash * (max(1,p ** (i - n + 1)) % MOD)) %
    MOD)
        if (currHash == (hash * (max(1,kpow[(i - n + 1)]) % MOD)) % MOD):
            print("YES")
```

```
            exit()
    print("NO")
```

Task E

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

const int maxn = 1e5+1;

int d[maxn] = {0};
vector <int> g[maxn];

int main() {
    int n, q;
    cin >> n >> q;
    for (int i = 1; i <= n; i++) {
        int cnt; cin >> cnt;
        g[i].resize(cnt);
        for (int j=0;j<cnt;j++) cin >> g[i][j];
    }
    vector <int> k(int(1e5+1)+1); k[0] = 1;
    for (int i = 1; i < int(1e5+1)+1; i++) {
        k[i] = g[k[i-1]][d[k[i-1]]];
        d[k[i-1]]++;
        d[k[i-1]] %= g[k[i-1]].size();
    }
    for (int i = 0; i < q; i++) {
        int x; cin >> x;
        cout << k[x] << " ";
    }
}
```

Task F

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

using namespace std;
using ll = long long int;

const int maxn = 200;

ll dp[maxn][maxn];
bool dpAns[maxn][maxn];
bool dpAns2[maxn][maxn];
bool dpAns3[maxn][maxn];
```

```
bool dpAns4[maxn][maxn];

int main() {
    int h, w;
    cin >> h >> w;
    for (int i = 0; i < h; i++)
        for (int j = 0; j < w; j++) {
            char x; cin >> x;
            dp[i][j] = ll(x - '0');
            dp[i+h][j] = dp[i][j];
            dp[i+h][j+w] = dp[i][j];
            dp[i][j+w] = dp[i][j];

        }

    ll ans = 0;;
    for (int i = 0; i < h; i += 2) {
        for (int j = 0; j < w; j += 2) {
            dpAns[i][j] = (dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][j
+1] > dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]);
            ans += max((dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][j
+1]), (dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]));
        }
    }
    ll ans2 = 0;
    for (int i = 0; i < h; i+=2) {
        for (int j = 1; j <= w; j+=2) {
            dpAns2[i][j] = (dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][
j+1] > dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]);
            ans2 += max((dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][j
+1]), (dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]));
        }
    }
    ll ans3 = 0;
    for (int i = 1; i <= h; i+=2) {
        for (int j = 0; j < w; j+=2) {
            dpAns3[i][j] = (dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][
j+1] > dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]);
            ans3 += max((dp[i][j] * dp[i+1][j] + dp[i][j+1] * dp[i+1][j
+1]), (dp[i][j]*dp[i][j+1] + dp[i+1][j]*dp[i+1][j+1]));
        }
    }

    if (ans > ans2 && ans > ans3) {
        cout << ans << "\n";
        for (int i = 0; i < h; i += 2) {
            for (int j = 0; j < w; j+=2) cout << (dpAns[i][j] ? "TT" : "
LR"); cout << "\n";
            for (int j = 0; j < w; j+=2) cout << (dpAns[i][j] ? "BB" : "
LR"); cout << "\n";
        }
    } else if (ans2 > ans3 && ans2 > ans) {
        cout << ans2 << "\n";
```

```
        for (int i = 0; i < h; i+= 2) {
            cout << (dpAns2[i][w-1] ? "T" : "R"); for (int j = 1; j <= w;
 j+=2) cout << (dpAns[i][j] ? "TT" : "LR"); cout << (dpAns2[i][w-1] ?
"T" : "L"); cout << "\n";
            cout << (dpAns2[i][w-1] ? "B" : "R"); for (int j = 1; j <= w;
 j+=2) cout << (dpAns[i][j] ? "BB" : "LR"); cout << (dpAns2[i][w-1] ?
"B" : "L"); cout << "\n";
        }
    } else {
        cout << ans3 << "\n";
        for (int j = 0; j < w; j += 2) {
            if (dpAns3[h][j]) cout << "BB";
            else cout << "LR";
        }
        cout << "\n";
        for (int i = 1; i <= h; i+=2) {
            for (int j = 0; j < w; j+=2) cout << (dpAns3[i][j] ? "TT" : "
LR"); cout << "\n";
            for (int j = 0; j < w; j+=2) cout << (dpAns3[i][j] ? "BB" : "
LR"); cout << "\n";
        }
        for (int j = 0; j < w; j += 2) {
            if (dpAns3[h][j]) cout << "TT";
            else cout << "LR";
        }
    }
}
```