杂题混练

人员

左子毅、朱奕鸣 到课

作业

https://vjudge.net/contest/658739, 上周作业题要求大家补完

课堂表现

这节课讲的几道题目都有些复杂，同学们课下一定要及时复习补题。

课堂内容

CF1898C Colorful Grid

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 16 + 5;
bool a[maxn][maxn], b[maxn][maxn];

void print(bool flag) { cout << (flag ? "R" : "B") << " "; }

void solve() {
    memset(a, 0, sizeof(a));
    memset(b, 0, sizeof(b));

    int n, m, k; cin >> n >> m >> k;
    int t = n+m-2;
    if (k<t || (k-t)&1) { cout << "NO" << endl; return; }

    cout << "YES" << endl;

    for (int i = 1; i <= m-1; i += 2) a[1][i] = true;
    for (int i = (m+1)%2+1; i <= n-1; i += 2) b[i][m] = b[i][m-1] = true;
    b[1][1] = b[1][2] = true;

    for (int i = 1; i <= n; ++i) {
        for (int j = 1; j <= m-1; ++j) print(a[i][j]);
        cout << endl;
    }

    for (int i = 1; i <= n-1; ++i) {
        for (int j = 1; j <= m; ++j) print(b[i][j]);
        cout << endl;
    }
}

int main()
{
    int T; cin >> T;
    while (T -- ) solve();
    return 0;
}
```

CF1902E Collapsing Strings

trie 树维护

```
#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int N = 1e6 + 5, M = 26;
int tr[N][M], f[N], id = 0;
string str[N];

void trInsert(string s) {
    int p = 0;
    for (char i : s) {
        int u = i - 'a';
        if (!tr[p][u]) tr[p][u] = ++id;
        p = tr[p][u]; ++f[p];
    }
}

LL trQuery(string s) {
    int p = 0; LL res = 0;
    for (char i : s) {
        int u = i - 'a';
        if (!tr[p][u]) break;
        p = tr[p][u]; res += f[p];
    }
    return res*2;
}

int main()
{
    int n; cin >> n;
    int tot = 0;
    for (int i = 1; i <= n; ++i) {
        cin >> str[i], tot += (int)str[i].size();
    }

    LL res = 0;
    for (int i = 1; i <= n; ++i) res += (LL)str[i].size()*n + tot;
    for (int i = 1; i <= n; ++i) trInsert(str[i]);
    for (int i = 1; i <= n; ++i) {
        string t = str[i]; reverse(t.begin(), t.end());
        res -= trQuery(t);
    }
    cout << res << endl;
    return 0;
}
```

CF1902D Robot Queries

```
#include <bits/stdc++.h>
#define x first
#define y second

using namespace std;

typedef pair<int, int> PII;
const int maxn = 2e5 + 5;
char s[maxn];
PII f[maxn];
map<PII, vector<int>> mp;
int dx[] = {0, 0, -1, 1}, dy[] = {1, -1, 0, 0};

int iValue(char x) {
  if (x == 'U') return 0;
  if (x == 'D') return 1;
  if (x == 'L') return 2;
  return 3;
}

void solve() {
  int x, y, l, r; cin >> x >> y >> l >> r;
  if (mp.count({x, y})) {
    int _l = mp[{x,y}][0], _r = mp[{x,y}].back();
    if (_l<l || _r>=r) { cout << "YES" << endl; return; }
  }

  int x1 = f[l-1].x, y1 = f[l-1].y, x2 = f[r].x, y2 = f[r].y;
  x = x1+x2-x, y = y1+y2-y;

  if (mp.count({x, y})) {
    vector<int>& vec = mp[{x,y}];
    vector<int>::iterator it = lower_bound(vec.begin(), vec.end(), l);
    if (it != vec.end() && *it <= r) { cout << "YES" << endl; return; }
  }

  cout << "NO" << endl;
}
```

```
int main()
{
  int n, T; cin >> n >> T;
  cin >> (s+1);

  mp[{0,0}].push_back(0); f[0] = {0, 0};
  for (int i = 1, x = 0, y = 0; i <= n; ++i) {
    int id = iValue(s[i]);
    x += dx[id], y += dy[id];
    mp[{x,y}].push_back(i);
    f[i] = {x, y};
  }

  while (T -- ) solve();
  return 0;
}
```