杂题混练

人员

蔡云翔、李佳声、胡赫轩、崔嘉睿 到课

作业

https://vjudge.net/contest/658739, 上周 5 道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

```cpp
#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, false, sizeof(a)), memset(b, false, 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 j = ((m-1)&1)+1; j <= n-1; j += 2) {
    b[j][m] = true;
    if (j+2 > n-1) b[j][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;
}
```

CF1904D2 Set To Max (Hard Version)

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

using namespace std;

void print(bool flag) { cout << (flag?"YES":"NO") << endl; }

const int N = 2e5 + 5, M = 20;
int a[N], b[N], w_log2[N];
int n;
vector<int> vec[N];
int f_max_a[N][M], f_min_b[N][M];

int query_max_a(int l, int r) {
  int k = w_log2[r-l+1];
  return max(f_max_a[l][k], f_max_a[r-(1<<k)+1][k]);
}

int query_min_b(int l, int r) {
  int k = w_log2[r-l+1];
  return min(f_min_b[l][k], f_min_b[r-(1<<k)+1][k]);
}

int lFind(int p) {
  int v = b[p];
  int lpos = lower_bound(vec[v].begin(), vec[v].end(), p) - vec[v].begin() - 1;
  if (lpos == -1) return -1;
  if (query_max_a(vec[v][lpos], p) > v) return -1;
  return vec[v][lpos];
}

int rFind(int p) {
```

```cpp
    int v = b[p];
    int rpos = lower_bound(vec[v].begin(), vec[v].end(), p) - vec[v].begin();
    if (rpos == (int)vec[v].size()) return -1;
    if (query_max_a(p, vec[v][rpos]) > v) return -1;
    return vec[v][rpos];
  }

  bool check(int l, int r, int v) {
    if (l==-1 || r==-1) return false;
    return query_min_b(l, r) >= v;
  }

  void solve() {
    cin >> n;
    for (int i = 1; i <= n; ++i) cin >> a[i];
    for (int i = 1; i <= n; ++i) cin >> b[i];

    for (int i = 1; i <= n; ++i) {
      if (a[i] > b[i]) return print(false);
    }

    for (int i = 1; i <= n; ++i) vec[i].clear();
    for (int i = 1; i <= n; ++i) {
      vec[a[i]].push_back(i);
      f_max_a[i][0] = a[i], f_min_b[i][0] = b[i];
    }
    for (int k = 1; k < M; ++k) {
      for (int i = 1; i+(1<<k)-1 <= n; ++i) {
        f_max_a[i][k] = max(f_max_a[i][k-1], f_max_a[i+(1<<(k-1))][k-1]);
        f_min_b[i][k] = min(f_min_b[i][k-1], f_min_b[i+(1<<(k-1))][k-1]);
      }
    }

    for (int i = 1; i <= n; ++i) {
      if (a[i] < b[i]) {
        int l = lFind(i), r = rFind(i);
        if (!check(l,i,b[i]) && !check(i,r,b[i])) return print(false);
      }
    }
    print(true);
  }

  int main()
  {
    for (int i = 0; (1<<i) < N; ++i) w_log2[1<<i] = i;
    for (int i = 2; i < N; ++i) {
      if (!w_log2[i]) w_log2[i] = w_log2[i-1];
    }

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

CF1905D Cyclic MEX

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

using namespace std;

typedef long long LL;
const int maxn = 2e6 + 5;
struct node {
  int l, r, v, maxx;
  LL sum;
} tr[maxn<<2];
int w[maxn], f[maxn];

void pushup(int u) {
  tr[u].sum = tr[u<<1].sum + tr[u<<1|1].sum;
  tr[u].maxx = max(tr[u<<1].maxx, tr[u<<1|1].maxx);
}

void pushdown(int u) {
  if (tr[u].v != -1) {
    tr[u<<1].v = tr[u].v, tr[u<<1].sum = (LL)tr[u<<1].v * (tr[u<<1].r-
tr[u<<1].l+1);
    tr[u<<1|1].v = tr[u].v, tr[u<<1|1].sum = (LL)tr[u<<1|1].v * (tr[u<<1|1].r-
tr[u<<1|1].l+1);
    tr[u<<1|1].maxx = tr[u].v, tr[u<<1|1].maxx = tr[u].v;
    tr[u].v = -1;
  }
}

void build(int u, int l, int r) {
  tr[u] = {l, r, -1, 0, 0};
  if (l == r) tr[u].maxx = tr[u].sum = f[l];
  else {
    int mid = (l + r) / 2;
    build(u<<1, l, mid), build(u<<1|1, mid+1, r);
    pushup(u);
  }
}

void modify(int u, int l, int r, int k) {
  if (tr[u].l>=l && tr[u].r<=r) {
    tr[u].v = tr[u].maxx = k, tr[u].sum = (LL)tr[u].v*(tr[u].r-tr[u].l+1);
    return;
  }
  pushdown(u);
  int mid = (tr[u].l + tr[u].r) / 2;
  if (l <= mid) modify(u<<1, l, r, k);
  if (r > mid) modify(u<<1|1, l, r, k);
  pushup(u);
}
```

```
LL query(int u, int l, int r) {
  if (tr[u].l>=l && tr[u].r<=r) return tr[u].sum;
  pushdown(u);
  int mid = (tr[u].l + tr[u].r) / 2;
  LL res = 0;
  if (l <= mid) res = query(u<<1, l, r);
  if (r > mid) res += query(u<<1|1, l, r);
  return res;
}

int queryPos(int u, int l, int r, int v) {
  if (tr[u].l == tr[u].r) return tr[u].l;
  if (tr[u<<1].maxx>v && l<=tr[u<<1].r) return queryPos(u<<1, l, r, v);
  return queryPos(u<<1|1, l, r, v);
}

void solve() {
  int n; cin >> n;
  for (int i = 1; i <= n; ++i) scanf("%d", &w[i]);

  set<int> s;
  for (int i = 0; i <= n; ++i) s.insert(i);
  for (int i = 1; i <= n; ++i) {
    s.erase(w[i]); f[i] = *s.begin();
  }

  build(1, 1, 2*n);

  LL res = 0;
  for (int i = 1; i <= n; ++i) {
    // 把 w[i] 移动到 w[n+i]
    // 在线段树 i+1 -- n+i-1 位置中找到第一个大于 w[i] 的位置 pos
    // 把 pos -- w[n+i-1] 的数都改成 w[i], 把 w[n+i] 的数改为 n
    // 对 i+1 -- n+i 进行求和即可
    int pos = queryPos(1, i+1, n+i-1, w[i]);
    modify(1, pos, n+i-1, w[i]); modify(1, n+i, n+i, n);
    res = max(res, query(1, i+1, n+i));
  }

  printf("%lld\n", res);
}

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