专题6

目录

1001 敌兵布阵

AC代码

```cpp
#include<bits/stdc++.h>
using namespace std;
int c[50010];
int n;
string s;
int lowbit(int x)
{
    return x & -x;
}
int sum(int i)
{
    int ret = 0;
    while (i > 0) {
        ret += c[i];
        i -= lowbit(i);
    }
    return ret;
}
void update(int i, int val)
{
    while (i <= n) {
        c[i] += val;
```

```
        i += lowbit(i);
    }
}
void solve()
{
    cin >> n;
    memset(c)
    for (int i = 1; i <= n; i++) {
        int temp;
        cin >> temp;
        update(i, temp);
    }
    while (cin >> s, s != "End") {
        if (s == "Query") {
            int a, b;
            cin >> a >> b;
            cout << sum(b) - sum(a - 1) << endl;
        }
        else if (s == "Add") {
            int a, plus;
            cin >> a >> plus;
            update(a, plus);
        }
        else if (s == "Sub") {
            int b, mus;
            cin >> b >> mus;
            update(b, -mus);
        }
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int tt;
    cin >> tt;
    for (int i = 1; i <= tt; i++) {
        cout << "Case " << i << ":" << endl;
        solve();
    }
    return 0;
}
```

1002 Color the ball

AC代码

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
int n;
int c[100010];
void add(int p, int x)
{
```

```
    while (p <= n)c[p] += x, p += p & -p;
}
void range_add(int l, int r, int x)
{
    add(l, x), add(r + 1, -x);
}
int ask(int p)
{
    int res = 0;
    while (p)res += c[p], p -= p & -p;
    return res;
}
void solve()
{
    memset(c, 0, sizeof(c));
    for (int i = 1; i <= n; i++) {
        int a, b;
        cin >> a >> b;
        range_add(a, b, 1);
    }
    for (int i = 1; i <= n-1; i++)
        cout << ask(i) << " ";
    cout << ask(n) << endl;
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    while(cin>>n,n)solve();
    return 0;
}
```

1003 Frosh Week

AC代码

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
int c[1000010];
int n;
struct node
{
    int num;
    int id;
}t[1000005];

bool cmp(const node& x, const node& y)
{
    return x.num < y.num;
}

int lowbit(int x)
{
```

```
    return x & -x;
}
ll sum(int i)
{
    ll ret = 0;
    while (i > 0) {
        ret += c[i];
        i -= lowbit(i);
    }
    return ret;
}
void update(int i, int val)
{
    while (i <= n) {
        c[i] += val;
        i += lowbit(i);
    }
}
void solve()
{
    while (cin >> n) {
        memset(c, 0, sizeof(c));
        ll su = 0;
        for (int i = 1; i <= n; i++) {
            cin >> t[i].num;
            t[i].id = i;
        }
        sort(t + 1, t + n + 1, cmp);
        for (int i = 1; i <= n; i++) {
            update(t[i].id, 1);
            su += (sum(n) - sum(t[i].id));
        }
        cout << su << endl;
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1004 See you~

AC代码

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
int c[1010][1010];
int mp[1010][1010];
int lowbit(int x){return x & -x;}
ll sum(int x,int y)
```

```
{
    ll ret = 0;
    int t;
    while (x > 0) {
        t = y;
        while (t > 0) {
            ret += c[x][t];
            t -= lowbit(t);
        }
        x -= lowbit(x);
    }
    return ret;
}
void update(int x, int y, int val)
{
    int t;
    while (x <= 1001) {
        t = y;
        while (t <= 1001) {
            c[x][t] += val;
            t += lowbit(t);
        }
        x += lowbit(x);
    }
}
void solve()
{
    memset(c, 0, sizeof(c));
    int q;
    int x1,x2,y1,y2, n1;
    cin >> q;
    for (int i = 1; i <= 1001; i++)
    {
        for (int j = 1; j <= 1001; j++)
        {
            mp[i][j] = 1;
            update(i, j, 1);
        }
    }
    while (q--) {
        string s;
        cin >> s;
        if (s == "S") {
            cin >> x1 >> y1 >> x2 >> y2;
            x1++, x2++, y1++, y2++;
            if (x1 > x2)swap(x1, x2);
            if (y1 > y2)swap(y1, y2);
            cout << sum(x2, y2) + sum(x1 - 1, y1 - 1) - sum(x1 - 1, y2) -
sum(x2, y1 - 1) << endl;
        }
        else if (s == "A") {
            cin >> x1 >> y1 >> n1;
            x1++, y1++;
            mp[x1][y1] += n1;
            update(x1, y1, n1);
        }
        else if (s == "D") {
            cin >> x1 >> y1 >> n1;
```

```
            x1++, y1++;
            if (mp[x1][y1] < n1) n1 = mp[x1][y1], mp[x1][y1] = 0;
            else mp[x1][y1] -= n1;
            update(x1, y1, -n1);
        }
        else {
            cin >> x1 >> y1 >> x2 >> y2 >> n1;
            x1++, y1++, x2++, y2++;
            if (mp[x1][y1] < n1) n1 = mp[x1][y1], mp[x1][y1] = 0;
            else mp[x1][y1] -= n1;
            mp[x2][y2] += n1;
            update(x1, y1, -n1);
            update(x2, y2, n1);
        }
    }

}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int tt;
    cin >> tt;
    for (int i = 1; i <= tt; i++) {
        cout << "Case " << i << ":"<<endl;
        solve();
    }
    return 0;
}
```

1005 Stars

AC代码

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

int n;
int level[32005];
int c[32005];
int sum(int x)
{
    int s = 0;
    while (x > 0)
    {
        s = s + c[x];
        x = x - (x & (-x));
    }
    return s;
}
void update(int x,int val)
{
    while (x <=32001){
        c[x] = c[x] + val;
```

```
        x = x + (x & (-x));
    }
}
void solve()
{
    memset(c, 0, sizeof(c));
    memset(level, 0, sizeof(level));
    for (int i = 0; i < n; i++)
    {
        int x, y;
        cin >> x >> y;
        x++;
        level[sum(x)]++;
        update(x,1);
    }
    for (int i = 0; i < n; i++)
        cout << level[i] << endl;
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    while (cin >> n)solve();
    return 0;
}
```

1006 I Hate It

AC代码

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

int lowerBit(int x) { return (x & (-x)); }
int num[200005], a[200005];
int n, m;
void Update(int x, int y)
{
    a[x] = y;
    while (x <= n)
    {
        num[x] = max(num[x], y);
        x += lowerBit(x);
    }
}
int ask(int x, int y)
{
    int res = a[y];
    while (x != y)
    {
        for (y -= 1; y - x >= lowerBit(y); y -= lowerBit(y))
        {
            res = max(res, num[y]);
        }
```

```
        res = max(res, a[y]);
    }
    return res;
}

void solve()
{
    while (cin>>n>>m){
        memset(num, 0, sizeof(num));
        for (int i = 1; i <= n; i++)
        {
            int t;
            cin >> t;
            Update(i, t);
        }
        string t;
        for (int i = 1; i <= m; i++)
        {
            cin >> t;
            if (t == "U")
            {
                int c, b;
                cin >> c >> b;
                Update(c, b);
            }
            if (t == "Q")
            {
                int c, b;
                cin >> c >> b;
                int ans = 0;
                cout << ask(c, b) << endl;
            }

        }
    }
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    solve();
    return 0;
}
```

1007 A Simple Problem with integers

AC代码

```
#include <bits/stdc++.h>
using namespace std;
const int MAXN = 50010;
int cl[MAXN][11][11], a[MAXN];
int n, q;

```

```
void add(int a, int k, int c) {
    int i = a, temp;
    while (i <= n) {
        temp = (i - a) % k;
        if (temp == 0) temp = k;
        cl[i][k][temp] += c;
        i += i & -i;
    }
}
int sum(int a) {
    int i = a;
    int ans = 0;
    while (i) {
        for (int j = 1; j < 11; j++)
            ans += cl[i][j][j - (a - i) % j];
        i -= i & -i;
    }
    return ans;
}
int main() {
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int k, l, r, c;
    while (cin>>n) {
        for (int i = 1; i <= n; i++)
            cin >> a[i];
        memset(cl, 0, sizeof(cl));
        cin >> q;
        while (q--) {
            int i;
            cin >> i;
            if (i == 1) {
                cin >> l >> r >> k >> c;
                add(l, k, c);
                add((r - l) / k * k + l + k, k, -c);
            }
            else {
                cin >> i;
                cout << sum(i) + a[i] << endl;
            }
        }
    }
    return 0;
}
```

1008 Just a Hook

AC代码

```
#include<bits/stdc++.h>
using namespace std;
const int maxn = 1e5 + 5;
typedef long long ll;
int sum[maxn << 2];
```

```cpp
int a[maxn << 2];

void pu(int rt)
{
    sum[rt] = sum[rt << 1] + sum[rt << 1 | 1];
}
void bd(int l, int r, int rt)
{
    a[rt] = 0;
    if (r == l) {
        sum[rt] = 1;
        return;
    }
    int m = (l + r) >> 1;
    bd(l, m, rt << 1);
    bd(m + 1, r, rt << 1 | 1);
    pu(rt);
}
void pd(int rt, int l, int r)
{
    if (a[rt]) {
        int m = (l + r) >> 1;
        a[rt << 1] = a[rt << 1 | 1] = a[rt];
        sum[rt << 1] = (m - l + 1) * a[rt];
        sum[rt << 1 | 1] = (r - m) * a[rt];
        a[rt] = 0;
    }
}
void update(int x, int y, int k, int l, int r, int rt)
{
    if (x <= l && y >= r) {
        a[rt] = k;
        sum[rt] = k * (r - l + 1);
        return;
    }
    int m = (l + r) >> 1;
    pd(rt, l, r);
    if (x <= m)update(x, y, k, l, m, rt << 1);
    if (m + 1 <= y)update(x, y, k, m + 1, r, rt << 1 | 1);
    pu(rt);
}
int qu(int x, int y, int l, int r, int rt)
{
    if (x <= 1 && y >= r)
        return sum[rt];
    pd(rt, l, r);
    int m = (l + r) >> 1;
    return qu(x, y, l, m, rt << 1) + qu(x, y, m + 1, r, rt << 1 | 1);
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int tt;
    int n, q;
    cin >> tt;
    int cnt = 0;
    while (tt--) {
```

```
        cin >> n >> q;
        int x, y, z;
        bd(1, n, 1);
        while (q--) {
            cin >> x >> y >> z;
            update(x, y, z, 1, n, 1);
        }
        cout << "Case " << ++cnt << ": The total value of the hook is ";
        cout << qu(1, n, 1, n, 1) << ".\n";
    }
    return 0;
}
```

1009 LCIS

AC代码

```
#include<bits/stdc++.h>
using namespace std;
typedef long long ll;
const int maxn = 1e5 + 5;

struct node {
    int lnum, rnum, lcnt, rcnt, ncnt;
}s[maxn<<2];
void pu(int rt, int m)
{
    s[rt].lnum = s[rt << 1].lnum;
    s[rt].rnum = s[rt << 1 | 1].rnum;
    s[rt].lcnt = s[rt << 1].lcnt;
    s[rt].rcnt = s[rt << 1 | 1].rcnt;
    s[rt].ncnt = max(s[rt << 1].ncnt, s[rt << 1 | 1].ncnt);

    if (s[rt << 1].rnum < s[rt << 1 | 1].lnum) {
        if (s[rt << 1].lcnt == m - (m >> 1))
            s[rt].lcnt += s[rt << 1 | 1].lcnt;
        if (s[rt << 1 | 1].rcnt == (m >> 1))
            s[rt].rcnt += s[rt << 1].rcnt;
        s[rt].ncnt = max(s[rt].ncnt, s[rt << 1].rcnt + s[rt << 1 | 1].lcnt);
    }
}
void bd(int l, int r, int rt)
{
    s[rt].lcnt = s[rt].rcnt = s[rt].ncnt = 1;
    if (l == r) {
        cin >> s[rt].lnum;
        s[rt].rnum = s[rt].lnum;
        return;
    }
    int m = (l + r) >> 1;
    bd(l, m, rt << 1);
    bd(m + 1, r, rt << 1 | 1);
    pu(rt, r - l + 1);
}
```

```cpp
void update(int id, int v, int l, int r, int rt)
{
    if (l == r) {
        s[rt].rnum = s[rt].lnum = v;
        s[rt].lcnt = s[rt].rcnt = s[rt].ncnt = 1;
        return;
    }
    int m = (l + r) >> 1;
    if (id <= m)update(id, v, l, m, rt << 1);
    if (id > m)update(id, v, m + 1, r, rt << 1 | 1);
    pu(rt, r - l + 1);
}
int qu(int L, int R, int l, int r, int rt)
{
    if (L <= l && R >= r)
        return s[rt].ncnt;
    int m = (l + r) >> 1;
    int ans = 0;
    if (L <= m)
        ans = max(ans, qu(L, R, l, m, rt << 1));
    if (R > m)
        ans = max(ans, qu(L, R, m + 1, r, rt << 1 | 1));
    if (s[rt << 1].rnum < s[rt << 1 | 1].lnum)
        ans = max(ans, min(m - L + 1, s[rt << 1].rcnt) + min(R - m, s[rt << 1 | 1].lcnt));
    return ans;
}
int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0); cout.tie(0);
    int tt;
    cin >> tt;
    while (tt--) {
        int n, q, x, y;
        cin >> n >> q;
        bd(1, n, 1);
        while (q--) {
            string s;
            cin >> s >> x >> y;
            if (s == "Q")cout << qu(x + 1, y + 1, 1, n, 1) << endl;
            if (s == "U")update(x + 1, y, 1, n, 1);
        }
    }
    return 0;
}
```

1010 Successor

未AC