

树状数组

人员

王毅博、褚锦轩、王承周、许睿谦、阮文璋 到课

上周作业检查

上周作业链接: <https://cppoj.kids123code.com/contest/1789>

比赛概况 题目列表 选择题列表 提交记录 ★ 实时榜单 ★ 选择题排行榜							
王向东老师周日一点半C++线段树练习							
刷新							
#	用户名	姓名	编程分	时间	A	B	C
1	wangchengzhou	王承周	300	3940	100	100	100
2	chujinxuan	褚锦轩	200	2409	100		100
3	ruanwenzhang	阮文璋	200	2623	100	100	
4	wangyibo	王毅博	200	2818	100		100
5	xuruiqian	许睿谦	100	1494	100	0	

本周作业

<https://cppoj.kids123code.com/contest/1911> (课上讲了 A ~ C 题 + 昨晚 abc F 题, 课后作业是 D 题)

课堂表现

今天课上讲了树状数组的内容, 这个内容比较简单, 同学们课上都吸收的比较好。

课上还讲了昨晚 abc 的 F 题, 这道题用树状数组可以比较轻松的完成, 同学们课下记得把这道题也补一补。

课堂内容

【模板】树状数组 1

树状数组模板, 单点修改, 区间查询

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

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
int tr[maxn];

int lowbit(int x) { return x & (-x); }

void update(int x, int k) {
    while (x < maxn) {
```

```
        tr[x] += k, x += lowbit(x);
    }
}

int query(int x) {
    int res = 0;
    while (x) {
        res += tr[x], x -= lowbit(x);
    }
    return res;
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x; update(i, x);
    }

    while (m -- ) {
        int op; cin >> op;
        if (op == 1) {
            int x, k; cin >> x >> k; update(x, k);
        } else {
            int l, r; cin >> l >> r;
            cout << query(r) - query(l-1) << endl;
        }
    }
    return 0;
}
```

【模板】树状数组 2

模板题: 树状数组 + 差分

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

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;
int tr[maxn];

int lowbit(int x) { return x & (-x); }

void update(int x, int k) {
    while (x < maxn) {
        tr[x] += k, x += lowbit(x);
    }
}
```

```

int query(int x) {
    int res = 0;
    while (x) {
        res += tr[x], x -= lowbit(x);
    }
    return res;
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1, last = 0; i <= n; ++i) {
        int x; cin >> x; update(i, x-last); last = x;
    }

    while (m -- ) {
        int op; cin >> op;
        if (op == 1) {
            int x, y, k; cin >> x >> y >> k;
            update(x, k), update(y+1, -k);
        } else {
            int x; cin >> x;
            cout << query(x) << endl;
        }
    }
    return 0;
}

```

逆序对

离散化 + 树状数组 查询 逆序对 数量

先把所有的数输入, 然后进行离散化

把每个数逐渐往树状数组中插入, 每次查询比 x 大的有多少

```

#include <bits/stdc++.h>

using namespace std;

vector<int> ys;
int yFind(int x) { return lower_bound(ys.begin(), ys.end(), x) - ys.begin(); }

typedef long long LL;
const int maxn = 5e5 + 5;
int w[maxn];

int tr[maxn];
int lowbit(int x) { return x & (-x); }
void update(int x, int k) {
    while (x < maxn) tr[x] += k, x += lowbit(x);
}

```

```

}
int query(int x) {
    int res = 0;
    while (x) res += tr[x], x -= lowbit(x);
    return res;
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i], ys.push_back(w[i]);
    sort(ys.begin(), ys.end()), ys.erase(unique(ys.begin(), ys.end()), ys.end());

    LL res = 0;
    for (int i = 1; i <= n; ++i) {
        int x = yFind(w[i])+1;
        res += (i-1) - query(x);
        update(x, 1);
    }
    cout << res << endl;
    return 0;
}

```

Starry Landscape Photo

题目链接: https://atcoder.jp/contests/abc436/tasks/abc436_f

考虑以 i 作为最大值输出时, 输出的情况数取决于左边有多少比 i 小的, 右边有多少比 i 大的, 这两个信息可以用树状数组进行维护

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 5e5 + 5;

int tr[maxn];
int lowbit(int x) { return x & (-x); }
void update(int x, int k) {
    while (x < maxn) tr[x] += k, x += lowbit(x);
}
int query(int x) {
    int res = 0;
    while (x) res += tr[x], x -= lowbit(x);
    return res;
}

int main()
{
    int n; cin >> n;

```

```
LL res = 0;
for (int i = 1; i <= n; ++i) {
    int x; cin >> x; update(x, 1);
    int num1 = query(x);
    int num2 = x - num1 + 1;
    res += (LL)num1 * num2;
}
cout << res << endl;
return 0;
}
```