树状数组

人员

刘佳赫、于珈浩 到课

作业

https://www.luogu.com.cn/contest/194725, A B C D 四道题

https://vjudge.net/contest/651049, 三道题

课堂表现

同学们课上基本把 树状数组 和 st表 都听懂了，但课下一定要再多复习，把这两个写熟，以后做题可以拿过来就用。

课堂内容

树状数组

```
// 树状数组: 单点修改, 区间查询 非常方便

// lowbit(x): x 的二进制中最后的一个 1

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

void update(int x, int k) {
    while (x <= n) {
        c[x] += k; x += lowbit(x);
    }
}

int query(int x) {
    int sum = 0;
    while (x) {
        sum += c[x]; x -= lowbit(x);
    }
    return sum;
}
```

T471235 graph

拆点, 将一个点拆成三个点，跑 bfs

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

```
using namespace std;

const int maxn = 1e5 + 5;
vector<int> vec[3*maxn];
int dis[3*maxn];

int f(int n, int k) { return 3*n + k; }

void add(int a, int b) { vec[a].push_back(b); }

int bfs(int st, int ed) {
  memset(dis, -1, sizeof(dis));
  queue<int> q; q.push(st); dis[st] = 0;
  while (!q.empty()) {
    int u = q.front(); q.pop();
    for (int i : vec[u]) {
      if (dis[i] == -1) {
        q.push(i); dis[i] = dis[u]+1;
      }
    }
  }

  return (dis[ed]==-1 ? -1 : dis[ed]/3);
}

int main()
{
  int n, m; cin >> n >> m;
  while (m -- ) {
    int a, b; cin >> a >> b;
    add(f(a,0), f(b,1));
    add(f(a,1), f(b,2));
    add(f(a,2), f(b,0));
  }
  int st, ed; cin >> st >> ed;
  cout << bfs(f(st,0), f(ed,0)) << endl;
  return 0;
}
```

P3374 【模板】树状数组 1

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

using namespace std;

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;
}
```

P3368 【模板】树状数组 2

1. 先修改为 差分数组, 然后 每次区间修改就只需要修改 2 个点了

2. 单点查询变为求前缀和

所以可以用 树状数组 解决

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

using namespace std;

const int maxn = 5e5 + 5;
int w[maxn], 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) cin >> w[i];
  for (int i = 1; i <= n; ++i) update(i, w[i] - w[i-1]);

  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;
}
```

P3865 【模板】ST 表 && RMQ 问题

ST 表模板题

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

using namespace std;

const int N = 1e5 + 5, M = 18;
int w[N], f[N][M], _log2[N];

void init(int n) {
    for (int k = 1; k < 18; ++k) {
        for (int i = 1; i+(1<<k)-1 <= n; ++i) {
            f[i][k] = max(f[i][k-1], f[i+(1<<(k-1))][k-1]);
        }
    }

    for (int i = 0; (1<<i) < N; ++i) _log2[(1<<i)] = i;
    for (int i = 1; i < N; ++i) {
        if (!_log2[i]) _log2[i] = _log2[i-1];
    }
}

int query(int l, int r) {
    int k = _log2[r-l+1];
    return max(f[l][k], f[r-(1<<k)+1][k]);
}

int main() {
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) cin >> w[i], f[i][0] = w[i];

    init(n);

    while (m -- ) {
        int l, r; scanf("%d%d", &l, &r);
        cout << query(l, r) << "\n";
    }
    return 0;
}
```