



南京工業大學
NANJING TECH
UNIVERSITY

ICPC Template Manual



作者: 贺梦杰

August 24, 2019

Contents

| | | |
|-----|-------------------------|---|
| 1 | 基础 | 2 |
| 2 | 搜索 | 3 |
| 3 | 动态规划 | 4 |
| 4 | 字符串 | 5 |
| 5 | 数据结构 | 6 |
| 5.1 | 带 Lazy 标记的线段树 | 7 |
| 6 | 图论 | 9 |

Chapter 1

基础

Chapter 2

搜索

Chapter 3

动态规划

Chapter 4

字符串

Chapter 5

数据结构

5.1 带 Lazy 标记的线段树

以下是区间修改 + 区间最大值查询。

若是区间修改 + 区间和查询，则在 pushdown 时需要将 lazy 标记乘上区间长度加到结点上。

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  const int N = 1e5 + 10;
5
6  int n; // 长度
7
8  struct Node {
9      int l, r, m;
10     int mx; // [l,r)中的最大值
11     int tag; // lazy标记
12 } t[4 * N]; // 数组大小不要忘记 * 4
13
14 // 将lazy标记下推
15 inline void pushdown(int x) {
16     Node& cur = t[x];
17     if (cur.r - cur.l == 1)
18         return;
19     Node &lch = t[x * 2], &rch = t[x * 2 + 1];
20     lch.mx += cur.tag, rch.mx += cur.tag;
21     lch.tag += cur.tag, rch.tag += cur.tag;
22     cur.tag = 0;
23 }
24
25 // 由x的儿子更新x结点，此时应确保x的儿子为最新
26 inline void pushup(int x) {
27     Node& cur = t[x];
28     if (cur.r - cur.l == 1)
29         return;
30     Node &lch = t[x * 2], &rch = t[x * 2 + 1];
31     cur.mx = max(lch.mx, rch.mx);
32 }
33
34 // 建树。注意初始时叶子是否为0，若不是，需要pushup
35 void build(int l, int r, int x) {
36     Node& cur = t[x];
37     cur.l = l, cur.r = r, cur.m = (l + r) / 2;
38     cur.mx = 0, cur.tag = 0; // 初始化最大值、lazy标记
39     if (r - l == 1)
40         return;
41     build(l, cur.m, x * 2);
42     build(cur.m, r, x * 2 + 1);
43     pushup(x); // 若初始值非0，则这句一定要加
44 }
45
46 // [l,r)每个元素加v。注意打标记时应更新被打标记结点的mx，并且应立即pushdown,pushup，过程中也需要pushdown,pushup
47 void update(int l, int r, int x, int v) {
48     Node& cur = t[x];
49     if (cur.l == l && cur.r == r) {
50         cur.tag += v, cur.mx += v; // 打标记的时候一定是同时更新mx的
51         pushdown(x), pushup(x);
52         return;
53     }
54     pushdown(x);
55     if (r <= cur.m)
56         update(l, r, x * 2, v);
57     else if (l >= cur.m)
58         update(l, r, x * 2 + 1, v);
59     else
60         update(l, cur.m, x * 2, v), update(cur.m, r, x * 2 + 1, v);
61     pushup(x);

```



```
62 }
63
64 // 查询[l,r)的最大值。过程中注意pushdown
65 int query(int l, int r, int x) {
66     Node& cur = t[x];
67     pushdown(x);
68     if (cur.l == l && cur.r == r)
69         return cur.mx;
70     int mx = 0;
71     if (r <= cur.m)
72         mx = query(l, r, x * 2);
73     else if (l >= cur.m)
74         mx = query(l, r, x * 2 + 1);
75     else
76         mx = max(query(l, cur.m, x * 2), query(cur.m, r, x * 2 + 1));
77     return mx;
78 }
79
80 int main() {
81     ios::sync_with_stdio(0);
82     cin.tie(0);
83
84     int m, i, l, r, v;
85     cin >> n >> m;
86     build(1, n + 1, 1); // 不要忘记build初始化, [l,r+1)
87     for (i = 1; i <= m; i++) {
88         cin >> v >> l >> r;
89         if (!v)
90             cout << query(l, r + 1, 1) << endl;
91         else
92             update(l, r + 1, 1, 1);
93     }
94
95     return 0;
96 }
```

Chapter 6

图论