

Nantong University ICPC Team Notebook (2018-19)

thirtiseven

目录

第一章	输入输出	2
1.1	取消同步	2
1.2	浮点数输出格式	2
第二章	数据结构	3
2.1	RMQ	3
2.2	线段树	3

第一章 输入输出

1.1 取消同步

```
1 std::ios::sync_with_stdio(false);  
2 std::cin.tie(0);
```

1.2 浮点数输出格式

```
1 //include <iomanip>  
2  
3 std::cout << std::fixed << std::setprecision(12) << ans << std::endl;
```

第二章 数据结构

2.1 RMQ

```
1  /*
2  RMQ模板:
3  A为原始数组, d[i][j]表示从i开始, 长度为(1<<j)的区间最小值
4  */
5  int A[maxn];
6  int d[maxn][30];
7
8  void init(int A[], int len) {
9      for (int i = 0; i < len; i++) d[i][0] = A[i];
10     for (int j = 1; (1 << j) <= len; j++) {
11         for (int i = 0; i + (1 << j) - 1 < len; i++) {
12             d[i][j] = min(d[i][j - 1], d[i + (1 << (j - 1))][j - 1]);
13         }
14     }
15 }
16
17 int query(int l, int r) {
18     int p = 0;
19     while ((1 << (p + 1)) <= r - l + 1) p++;
20     return min(d[l][p], d[r - (1 << p) + 1][p]);
21 }
```

2.2 线段树

```
1  /*
2  线段树模板:
3  A为原始数组, sum记录区间和, Add为懒惰标记
4  */
5
6  int A[maxn], sum[maxn << 2], Add[maxn << 2];
7
8  void pushup(int rt) {
9      sum[rt] = sum[rt << 1] + sum[rt << 1 | 1];
10 }
11
12 void pushdown(int rt, int l, int r) {
13     if (Add[rt]) {
```

```
14     int mid = (l + r) >> 1;
15     Add[rt << 1] += Add[rt];
16     Add[rt << 1 | 1] += Add[rt];
17     sum[rt << 1] += (mid - l + 1)*Add[rt];
18     sum[rt << 1 | 1] += (r - mid)*Add[rt];
19     Add[rt] = 0;
20 }
21 }
22
23 void build(int l, int r, int rt) {
24     if (l == r) {
25         sum[rt] = A[l];
26         return;
27     }
28     int mid = (l + r) >> 1;
29     build(l, mid, rt << 1);
30     build(mid + 1, r, rt << 1 | 1);
31     pushup(rt);
32 }
33
34 // 区间加值
35 void update(int L, int R, int val, int l, int r, int rt) {
36     if (L <= l && R >= r) {
37         Add[rt] += val;
38         sum[rt] += (r - l + 1)*val;
39         return;
40     }
41     pushdown(rt, l, r);
42     int mid = (l + r) >> 1;
43     if (L <= mid)update(L, R, val, l, mid, rt << 1);
44     if (R > mid)update(L, R, val, mid + 1, r, rt << 1 | 1);
45     pushup(rt);
46 }
47
48 // 点修改
49 void update(int index, int val, int l, int r, int rt) {
50     if (l == r) {
51         sum[rt] = val;
52         return;
53     }
54     int mid = (l + r) >> 1;
55     if (index <= mid)update(index, val, l, mid, rt << 1);
56     else update(index, val, mid + 1, r, rt << 1 | 1);
57     pushup(rt);
58 }
59
60 // 区间查询
61 int query(int L, int R, int l, int r, int rt) {
62     if (L <= l && R >= r) {
63         return sum[rt];
64     }
65     pushdown(rt, l, r);
66     int mid = (l + r) >> 1;
67     int ret = 0;
```

```
68     if (L <= mid)ret += query(L, R, l, mid, rt << 1);
69     if (R > mid)ret += query(L, R, mid + 1, r, rt << 1 | 1);
70     return ret;
71 }
```
