南京工業大學
NANJING TECH UNIVERSITY

ICPC Template Manual

作者: 贺梦杰

August 24, 2019

Contents

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 划分树

5.1.1 简介

划分树是模拟快速排序的。快速排序，自顶向下越来越有序。划分树是在进行快速排序的过程中记录当前子段中每个位置的数是否进入左子树，并用前缀和的思想统计它们。之后再查询时便可依据进入左子树的个数计算第 k 个数是在左子树还是右子树。

5.1.2 区间第 k 小值

5.1.2.1 问题简述

给定 n 个数，m 次查询，每次查询 [l,r] 内从小到大第 k 个数，输出这个数。

5.1.2.2 解决方案

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

using namespace std;
typedef long long ll;
const int N = 1e5 + 10;

int a[N], sorted[N], n;
int seg[21][N], toleft[21][N];

// 建树
void build(int l, int r, int dep) {
    if (l == r) {
        seg[dep + 1][l] = seg[dep][l];
        return;
    }
    int i, m = (l + r) / 2, lp = l, rp = m + 1, cnt = 0, t = 0;
    // cnt: 该节点内比sorted[m]小的元素个数
    for (i = l; i <= r; i++)
        cnt += seg[dep][i] < sorted[m];

    for (i = l; i <= r; i++) {
        if (i == l)
            toleft[dep][i] = 0;
        else
            toleft[dep][i] = toleft[dep][i - 1];

        if (seg[dep][i] < sorted[m])
            seg[dep + 1][lp++] = seg[dep][i], toleft[dep][i]++;
        else if (seg[dep][i] > sorted[m])
            seg[dep + 1][rp++] = seg[dep][i];
        else {                          // ==
            if (t < m - l + 1 - cnt)  // m-l+1-cnt: 左边最多可以放多少个sorted[m]
                seg[dep + 1][lp++] = seg[dep][i], toleft[dep][i]++, t++;
            else
                seg[dep + 1][rp++] = seg[dep][i];
        }
    }

    build(l, m, dep + 1);
    build(m + 1, r, dep + 1);
}

// 询问区间[l,r]内从小到大的第k个值，当前区间为[sl,sr](初始时为[1,n])，当前深度为dep(初始时为1)
int query(int l, int r, int k, int sl, int sr, int dep) {
    if (r == l)
        return seg[dep][l];
    int m = (sl + sr) / 2;
    int tlsl = (l - 1 >= sl ? toleft[dep][l - 1] : 0), tlsr = toleft[dep][sr];
    int tlr = toleft[dep][r];
    if (k <= tlr - tlsl)
```

```
51          return query(sl + tlsl, m - (tlsr - tlr), k, sl, m, dep + 1);
52      else
53          return query(m + 1 + l - sl - tlsl, sr - (sr - r - (tlsr - tlr)), k - (tlr - tlsl), m +
        1, sr, dep + 1);
54  }
55
56  int main() {
57      int m, i, l, r, k;
58      scanf("%d%d", &n, &m);
59
60      // 以下4行是初始化
61      for (i = 1; i <= n; i++)
62          scanf("%d", &a[i]), sorted[i] = seg[1][i] = a[i];
63      sort(sorted + 1, sorted + 1 + n);
64      build(1, n, 1);
65
66      for (i = 1; i <= m; i++) {
67          scanf("%d%d%d", &l, &r, &k);
68          int ans = query(l, r, k, 1, n, 1);
69          printf("%d\n", ans);
70      }
71
72      return 0;
73  }
```

Chapter 6

图论