



南京工業大學  
NANJING TECH  
UNIVERSITY

## ICPC Template Manual



作者: 贺梦杰

September 18, 2019

# Contents

<b>1</b>	<b>基础</b>	<b>2</b>
1.1	测试 . . . . .	3
<b>2</b>	<b>搜索</b>	<b>4</b>
<b>3</b>	<b>动态规划</b>	<b>5</b>
<b>4</b>	<b>字符串</b>	<b>6</b>
<b>5</b>	<b>数据结构</b>	<b>7</b>
5.1	区间最大连续子段和 . . . . .	8
<b>6</b>	<b>图论</b>	<b>10</b>

# Chapter 1

## 基础

## 1.1 测试

## Chapter 2

# 搜索

## Chapter 3

# 动态规划

## Chapter 4

# 字符串

## Chapter 5

# 数据结构



## 5.1 区间最大连续子段和

```

1  #include <bits/stdc++.h>
2
3  using namespace std;
4  const int N = 5e4 + 10; // 数组大小, 记得改
5
6  struct Node {
7      int l, r, m;
8      int s, f, fl, fr; // 区间的和, 区间最大子段和, 包含左端点的最大子段和, 包含右端点的最大子段和
9  } s[N * 4];
10
11 // 构建线段树
12 void build(int l, int r, int i) {
13     Node& fa = s[i];
14     fa.l = l, fa.r = r, fa.m = (l + r) / 2;
15     fa.s = fa.f = fa.fl = fa.fr = 0;
16     if (r - l == 1)
17         return;
18     build(l, fa.m, i * 2);
19     build(fa.m, r, i * 2 + 1);
20 }
21
22 // 自底向上更新
23 void pushup(int i) {
24     Node &fa = s[i], &lson = s[i * 2], &rson = s[i * 2 + 1]; // 父亲 左儿子 右儿子
25     fa.s = lson.s + rson.s;
26     fa.f = max(max(lson.f, rson.f), lson.fr + rson.fl);
27     fa.fl = max(lson.fl, lson.s + rson.fl);
28     fa.fr = max(rson.fr, rson.s + lson.fr);
29 }
30
31 // 单点更新
32 void update(int x, int p, int i) {
33     Node& fa = s[i];
34     if (fa.l == p && fa.r - fa.l == 1) {
35         fa.s = fa.f = fa.fl = fa.fr = x;
36         return;
37     }
38     if (p < fa.m)
39         update(x, p, i * 2);
40     else
41         update(x, p, i * 2 + 1);
42     // 向上更新
43     pushup(i);
44 }
45
46 // 作为查询的返回值
47 struct Ret {
48     int f, s, fl, fr;
49 };
50
51 // 查询
52 Ret query(int l, int r, int i) {
53     Node& fa = s[i];
54     if (fa.l == l && fa.r == r)
55         return {fa.f, fa.s, fa.fl, fa.fr};
56     if (r <= fa.m)
57         return query(l, r, i * 2);
58     else if (l >= fa.m)
59         return query(l, r, i * 2 + 1);
60     else {
61         Ret lret = query(l, fa.m, i * 2);
62         Ret rret = query(fa.m, r, i * 2 + 1);
63         return {max(max(lret.f, rret.f), lret.fr + rret.fl), lret.s + rret.s, max(lret.fl, lret.s
+ rret.fl), max(rret.fr, rret.s + lret.fr)};

```

```
64     }
65 }
66
67 int main() {
68     ios::sync_with_stdio(0);
69     cin.tie(0);
70
71     int n, m, i, x, l, r;
72     cin >> n;
73     build(1, n + 1, 1); // 不要忘了初始化
74     for (i = 1; i <= n; i++) {
75         cin >> x;
76         update(x, i, 1); // 修改元素
77     }
78     cin >> m; // 询问
79     while (m--) {
80         cin >> l >> r;
81         cout << query(1, r + 1, 1).f << endl; // 询问[1,r], 实际查询[1,r+1), 统一用左闭右开区间
82     }
83
84     return 0;
85 }
```

## Chapter 6

# 图论