

简单易懂的现代魔法 数据结构

郭晓旭

上海交通大学

2015 年 4 月 15 日

Prerequisite

假定观众具有基本的数据结构知识

Best pair

- 数列 a_1, a_2, \dots, a_n
- 求 $\min\{a_i, a_j\} \times |j - i|$ 的最大值
- $a_i \geq 0$
- $O(n \log n)$ 是不可以的

每次删掉小的一头
正确性？

每次删掉小的一头
正确性？
想不到怎么办？



Median

- 数列 a_1, a_2, \dots, a_n
- 求所有的 $median(\{a_l, a_{l+1}, \dots, a_r\})$
- $O(n^2 \log n)$ 是不可以的

删除比插入简单

固定 l , 依次计算 $r = n, n-1, \dots, l$ 时的答案

排序 $\{a_l, a_{l+1}, \dots, a_n\}$, 用链表维护

考虑删除元素后中位数的变化

删除比插入简单

固定 l , 依次计算 $r = n, n - 1, \dots, l$ 时的答案

排序 $\{a_l, a_{l+1}, \dots, a_n\}$, 用链表维护

考虑删除元素后中位数的变化

排序?

Shortest

- n 个点 m 条边的无向图
- 边的长度是 1 或 2
- 求点 1 到点 n 的最短路

❶ $2 = 1 + 1$

❷ 长度是 $O(n)$ 的, $O(n)$ 个 `std::vector`

Away

- n 行 n 列的点阵, k 个特殊点 (x_i, y_i)
- 安全性 $w(a, b) = \min_i \{|a - x_i| + |b - y_i|\}$
- 求一条从 $(1, 1)$ 到 (n, n) 的路径
- 最大化路径上安全性的最小值
- $O(n^2 \log n)$ 是不可以的

计算 $w(a, b)$

就是最短路嘛！

不二分最小值怎么办呢？

从大到小枚举安全性的最小值
最短路就是发大水，现在大水退去……
用并查集维护陆地的连通性

用 $\min\{d(u), w(v)\}$ 更新 $d(v)$ 的最短路
链表当然是支持辣!



用 $\min\{d(u), w(v)\}$ 更新 $d(v)$ 的最短路
链表当然是支持辣!



注意到如果 u, v 相邻, 那么 $|w(u) - w(v)| \leq 1$
换言之, 队里里面只有 x 和 $(x-1)$
要么塞到头, 要么塞到尾 XD

Editor¹

进行 q 次操作

insert(x) (在光标后) 插入数字 x

delete (在光标前) 删除数字

left/right 向左 (右) 移动光标

query(k) 设光标前的序列是 $\{a_1, a_2, \dots, a_n\}$,

$s_i = a_1 + a_2 + \dots + a_i$, 求 s_1, s_2, \dots, s_k 的最大值

$O(q \log q)$ 是不可以的

¹<http://acm.hdu.edu.cn/showproblem.php?pid=4699>

从光标处劈开，用两个栈分别维护

树状数组

```
void add(int k, int v) {  
    for (; k <= n; k += -k & k) {  
        a[k] += v;  
    }  
}
```

```
int ask(int k) {  
    int ret = 0;  
    for (; k; k -= -k & k) {  
        ret += a[k];  
    }  
    return ret;  
}
```

a[0] 的存在感

```
void add(int k, int v) {  
    for (; k < n; k += ~k & k + 1) {  
        a[k] += v;  
    }  
}
```

```
int ask(int k) {  
    int ret = 0;  
    for (; ~k; k -= ~k & k + 1) {  
        ret += a[k];  
    }  
    return ret;  
}
```

线段树

$2n, 3n$ 还是 $4n$ 空间?

线段树

$2n, 3n$ 还是 $4n$ 空间?

把区间 $[l, r]$ 存在 `get_id(l, r)`

```
int get_id(l, r) {  
    return l + r | l != r;  
}
```

只有 push_down 和 update 的世界

```
void insert(int l, int r, int a, int b) {  
    if (a <= l && r <= b) {  
        // modify  
    } else if (l <= b && a <= r) {  
        push_down(l, r);  
        int m = l + r >> 1;  
        insert(l, m, a, b);  
        insert(m + 1, r, a, b);  
        update(l, r);  
    }  
}
```

Treap

```
void rotate(int &x, int t) {  
    int y = children[x][t];  
    children[x][t] = children[y][1 ^ t];  
    children[y][1 ^ t] = x;  
    update(x);  
    update(y);  
    x = y;  
}
```

```
weight[0] = RAND_MAX;

void insert(int &x, int k) {
    if (!x) {
        x = count ++;
        key[x] = k;
        weight[x] = rand();
    } else {
        int t = key[x] < k;
        insert(children[x][t], k);
        if (weight[children[x][t]] < weight[x]) {
            rotate(x, t);
        }
    }
    update(x);
}
```



```

void erase(int &x, int k) {
    if (key[x] != k) {
        erase(children[x][key[x] < k], k);
    } else if (!children[x][0] && !children[x][1]) {
        x = 0;
        return;
    } else {
        rotate(x,
            weight[children[x][0]] > weight[children[x][1]]);
        erase(x, k);
    }
    update(x);
}

```

CVS²

进行 q 次操作

learn(i, j) i 学会了 j

rollback(i) i 失忆惹

relearn(i) i 又想起来惹

clone(i) 产生了 i 的克隆人

check(i) i 最近学了啥?

$O(q)$

²<http://codeforces.com/gym/100285/problem/C>

当然是 (2 个) 持久化栈
可是怎么实现呢？



GRE Words Revenge³

维护字典 D , 支持:

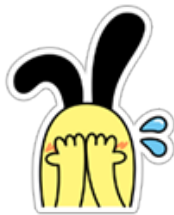
insert(s) 在字典 D 中插入字符串 s

query(t) 询问 t 有多少个子串在字典 D 中

$|D| \leq 10^5$, **强制在线**

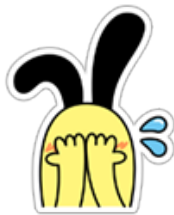
³<http://acm.hdu.edu.cn/showproblem.php?pid=4787>

一个动态的 AC 自动机？



关键是 `fail[]` 不好维护

一个动态的 AC 自动机？



关键是 `fail[]` 不好维护

维护**大**自动机 B 和**小**自动机 S ，保证 $|S| \leq \sqrt{|D|}$

每次优先插入到 S ，重新计算 S 的 `fail[]`

一旦 S 爆炸，重建 B ，然而只会爆炸 $\sqrt{|D|}$ 次

GRE Words Once More!⁴

- 无环的确定有限状态自动机 G
- 询问字典序第 k 小的字符串的长度
- $|G| \leq 10^5, k \leq 10^8$

⁴<http://acm.hdu.edu.cn/showproblem.php?pid=5118>

自然的问题：如果我手上有后缀自动机，如果得到对应的后缀数组？

$s(u)$ 表示从 u 出发的所有字符串，按照字典序排序
则

$$s(u) = \bigcup_{c \in \Sigma} (c + s(\text{go}(u, c)))$$

看上去是持久化链表



持久化 Treap

```
bool random(int p, int q) {  
    return rand() % (p + q) < p;  
}
```

```
Node* merge(Node *a, Node* b) {  
    if (a == null) return b;  
    if (b == null) return a;  
    if (random(a->size, b->size)) {  
        return new Node(a->left, merge(a->right, b));  
    } else {  
        return new Node(merge(a, b->left), merge(b));  
    }  
}
```

```
typedef std::pair <Node*, Node*> Pair;

Pair split(Node *a, int n) {
    if (a == null) return std::make_pair(null, null);
    if (n <= a->left->size) {
        Pair p = split(a->left, n);
        return std::make_pair(p.first,
                               new Node(p.second, a->right));
    } else {
        Pair p = split(a->right, n - 1 - a->left->size);
        return std::make_pair(
            new Node(a->left, p.first), p.second);
    }
}
```

让我们忘记 Splay 的悲伤……

喜鹊⁵

- n 个 3 元组 $(a_1, b_1, c_1), (a_2, b_2, c_2), \dots, (a_n, b_n, c_n)$
- 求 X, Y, Z 满足 $X \cup Y \cup Z = \{1, 2, \dots, n\}$
- 最小化 $\max_{i \in X} a_i + \max_{i \in Y} b_i + \max_{i \in Z} c_i$
- $a_i, b_i, c_i \geq 0$
- $O(n \log n)$

⁵<http://www.rqnoj.cn/problem/469>

Heavy-Light Decomposition⁶

- n 个节点的二叉树
- 删除叶子
- 维护轻重边剖分中重链的权值和
- $O(n \log n)$

⁶<http://codeforces.com/gym/100524/problem/H>, by Andrew Stankevich

谢谢大家