

# 树状数组

## 树状数组

### 一、前缀和与差分

问题1：原数组区间和操作

问题2：原数组区间元素修改（加法）

### 二、树状数组

### 三、海贼 OJ-329-弱化的整数问题

引入差分数组

结论

### 四、海贼 OJ-330-加强的整数问题

引入差分数组

原数组上的区间和问题转化

结论

## 一、前缀和与差分

---

1. 原数组：  $a_1, a_2, a_3, \dots, a_n$
2. 前缀和：  $S_i = \sum_{k=1}^i a_k$ ,  $a_i = S_i - S_{i-1}$
3. 差分数组：  $X_i = a_i - a_{i-1}$
4. X 数组是 a 数组的差分数组，a 数组是 S 数组的差分数组
5. S 数组是 a 数组的前缀和数组，a 数组是 X 数组的前缀和数组
6. 前缀和数组以及差分数组，并没有增加信息，只是信息的另外一种表示形式
7. 前缀和数组用来优化区间和操作
8. 差分数组用来优化区间修改操作

### 问题1：原数组区间和操作

a 数组上的操作：  $O(n)$

S 数组上的操作：  $O(1)$ ,  $S_i - S_{j-1} = a[j, i]$  区间和

### 问题2：原数组区间元素修改（加法）

$a = \{a_1, a_2, a_3, a_4, a_5, a_6\}$

$X = \{X_1 = a_1 - a_0, X_2 = a_2 - a_1, X_3 = a_3 - a_2, X_4 = a_4 - a_3, X_5 = a_5 - a_4, X_6 = a_6 - a_5\}$

$$\{a_1, a_2 + d, a_3 + d, a_4 + d, a_5, a_6\}$$

$$X = \{X_1 = a_1 - a_0, X_2 = a_2 - a_1 + d, X_3 = a_3 - a_2, X_4 = a_4 - a_3, X_5 = a_5 - a_4 - d, X_6 = a_6 - a_5\}$$

a 数组时间复杂度:  $O(n)$

x 数组时间复杂度:  $O(1)$

**树状数组**

一、前缀和数组

前缀和  $\rightarrow S_0=0 \quad S_1=a_1 \quad S_2=S_1+a_2 \quad S_3=S_2+a_3$

原数组  $\rightarrow a_0=0 \quad a_1 \quad a_2 \quad a_3 \quad a_4 \quad a_5 \quad a_6$

差分数组  $\rightarrow X = \quad \underline{X_1=a_1-a_0 \quad X_2=a_2-a_1 \quad X_3=a_3-a_2}$

↑

## 二、树状数组

1. lowbit 函数求数字  $i$ , 二进制表示中的最低1所在的位权
2.  $\text{lowbit}(x) = x \& -x$
3. 树状数组本质上是对前缀和数组的一种优化, 主要体现在单点修改操作上
4. 前缀和查询  $O(\log n)$ , 单点修改  $O(\log n)$
5. 相比于最普通的前缀和数组, 查询方面变差, 单点修改操作变好, 综合时间复杂度变好
6. 查询的时候, 向前统计,  $i$  的前一位  $i - \text{lowbit}(i)$
7. 修改的时候, 向后修改,  $i$  的后一位  $i + \text{lowbit}(i)$

## 三、海贼 OJ-329-弱化的整数问题

## 引入差分数组

$$a = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

$$X = \{X_1 = a_1 - a_0, X_2 = a_2 - a_1, X_3 = a_3 - a_2, X_4 = a_4 - a_3, X_5 = a_5 - a_4, X_6 = a_6 - a_5\}$$

$$\{a_1, a_2 + d, a_3 + d, a_4 + d, a_5, a_6\}$$

$$X = \{X_1 = a_1 - a_0, X_2 = a_2 - a_1 + d, X_3 = a_3 - a_2, X_4 = a_4 - a_3, X_5 = a_5 - a_4 - d, X_6 = a_6 - a_5\}$$

引入差分数组  $X$ ，将原数组  $a$  上的区间加操作，转换成  $X$  数组上的两次【单点操作】

对于查询原数组  $a[i]$  的值，等价于查询  $X$  数组前  $i$  位的【前缀和】

## 结论

由于，既要维护【前缀和】，又要进行【单点修改】，所以可以使用树状数组

**#329. 弱化的整数问题**

**题目描述**

给定长度为  $N$  ( $N \leq 100000$ ) 的序列  $A$  ( $0 \leq A_i \leq 100000$ )，然后输入  $m$  ( $m \leq 10^5$ ) 行操作指令

第一类指令形如  $C\ l\ r\ d$  ( $1 \leq l \leq r \leq N$ )，表示把数列中第  $l \dots r$  之间的数都加  $d$  ( $0 \leq d \leq 100000$ )

第二类指令形如  $Q\ x$  ( $x \leq N$ )，表示询问序列中第  $x$  个数的值。

**输入**

第一行一个整数  $N$ ，代表序列  $A$  的长度

第二行是由空格分隔开的  $N$  个数，分别代表  $A_1, A_2, \dots, A_n$

接下来一行是一个整数  $m$ ，代表操作的次数。

接下来  $m$  行，每行代表这一条指令如题目所述

**输出**

对于每次  $Q$  查询，输出一行为查询的值。

**输入样例1**

5

## 四、海贼 OJ-330-加强的整数问题

## 引入差分数组

参考 HZOJ-329 的解法，主要为了维护原数组上的区间修改操作

## 原数组上的区间和问题转化

$$a = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

$$X = \{X_1 = a_1 - a_0, X_2 = a_2 - a_1, X_3 = a_3 - a_2, X_4 = a_4 - a_3, X_5 = a_5 - a_4, X_6 = a_6 - a_5\}$$

$Query(l, r) = S(r) - S(l - 1)$ ，重点分析  $S$  怎么求，会求  $S$ ，万事大吉

$$S_i = \sum_{k=1}^i \sum_{y=1}^k X_y = \sum_{k=1}^i i(i+1)X_k - k * X_k = (i+1) \sum_{k=1}^i X_k - \sum_{k=1}^i k * X_k$$

$$\text{设 } Y_i = i \times X_i$$

$$S_i = (i+1) \sum_{k=1}^i X_k - \sum_{k=1}^i Y_k$$

## 结论

$S_i$  可以通过维护  $X$  与  $Y$  两个序列的前缀和得到

所以可以通过维护两个与差分数组  $X$  相关的前缀和数组，从而得到原数组  $a$  的前缀和值

需要维护两个：树状数组

The image shows a handwritten derivation of the formula for  $S_i$  using difference arrays  $X$  and  $Y$ . At the top, a diagram shows an array  $X$  with indices  $i$  and  $j$  marked. To the right, a box contains the conclusion:  $G_1 \rightarrow X$  前缀和,  $G_2 \rightarrow Y$  前缀和, and  $Y_k = X_k \cdot k$ . Below this, the formula for  $Query(i, j) = S_j - S_{j-1}$  is written. The main derivation shows  $S_i = \sum_{k=1}^i \sum_{y=1}^k X_y = \sum_{k=1}^i (i-k+1)X_k = (i+1) \sum_{k=1}^i X_k - \sum_{k=1}^i kX_k$ . At the bottom, a diagram illustrates the expansion of the sum for  $k=1, 2, 3$ , showing how the terms are grouped to match the final formula. The final formula is also written again on the right:  $(i+1) \sum_{k=1}^i X_k - \sum_{k=1}^i Y_k$ .

