树状数组

胡船长

初航我带你，远航靠自己

一、前缀和与差分

1. 一维前缀和
2. 二维前缀和
3. 差分数组的应用

二、树状数组

1. 树状数组基础知识
2. 树状数组-课后练习题

一、前缀和与差分

1. 一维前缀和
2. 二维前缀和
3. 差分数组的应用

前　缀　和

定义:

A 序列第 i 个位置的【前缀和】定义为: $S_i = \sum_{k=1}^{i} A_i$

其中 S_i 是 S 序列的第 i 个位置的元素，特殊的: $S_0 = 0$

举例:

A 序列:		4	2	6	5	1	3	2
S 序列:	0	4	6	12	17	18	21	23

前 缀 和

性质:

A 序列的区间和: O(n) 的操作

S 序列的两项差: O(1) 的操作

举例:

A 序列:		4	2	6	5	1	3	2
S 序列:	0	4	6	12	17	18	21	23

A[1~5] = 18

S[5]-S[0] = 18

A[i~j] = S[j] - S[i-1]

前缀和-例题

1. B3612：求区间和
2. P1115：最大子段和

一、前缀和与差分

1. 一维前缀和
2. 二维前缀和
3. 差分数组的应用

二维前缀和

定义:

M 矩阵 (i,j) 位置的【二维前缀和】定义为: $G_{i,j}=\sum_{k=1}^{i}\sum_{l=1}^{j}M_{k,l}$

举例:

M 矩阵

	5	3	1	3
	6	2	4	2
	7	8	3	6
	9	2	4	9

G 矩阵

0	0	0	0	0
0	5	8	9	12
0	11	16	21	26
0	18	31	39	50
0	27	42	54	74

二维前缀和

性质:

M 矩阵的区域和: O(n*m) 的操作

G 矩阵的四项值: O(1) 的操作

举例:

结果 = 蓝 − 红 − 绿 + 紫

M 矩阵

	5	3	1	3
	6	2	4	2
	7	8	3	6
	9	2	4	9

G 矩阵

0	0	0	0	0
0	5	8	9	12
0	11	16	21	26
0	18	31	39	50
0	27	42	54	74

HZOJ-232：激光炸弹

二维前缀和 $O(n^2)$：

1、初始化二维前缀和数组G，统计得到每个点左上方区域的目标总价值

2、利用二维前缀和数组G，加速求解二维区域内的价值和值。

一、前缀和与差分

1. 一维前缀和
2. 二维前缀和
3. 差分数组的应用

差 分 数 组

定义:

A 序列第 i 个位置的【差分】定义为：$X_i = A_{\mathrm{i}} - A_{\mathrm{i-1}}$

其中 X_i 是 X 序列的第 i 个位置的元素，特殊的：$X_1 = A_1$

举例:

A 序列:	4	2	6	5	1	3	2
X 序列:	4	-2	4	-1	-4	2	-1

差 分 数 组

定义:

A 序列第 i 个位置的【差分】定义为：$X_i = A_{\mathrm{i}} - A_{\mathrm{i-1}}$

其中 X_i 是 X 序列的第 i 个位置的元素，特殊的：$X_1 = A_1$

举例:

A 序列:		4	2	6	5	1	3	2
X 序列:		4	-2	4	-1	-4	2	-1
S 序列:	0	4	2	6	5	1	3	2

差分数组

定义:

A 序列第 i 个位置的【差分】定义为：$X_i = A_{\mathrm{i}} - A_{\mathrm{i-1}}$

其中 X_i 是 X 序列的第 i 个位置的元素，特殊的：$X_1 = A_1$

前缀和与差分，互为逆运算

举例:

A 序列:		4	2	6	5	1	3	2
X 序列:		4	-2	4	-1	-4	2	-1
S 序列:	0	4	2	6	5	1	3	2

差 分 数 组

性质:

A 序列的区间 +d: O(n) 的操作

X 序列的2点操作: O(1) 的操作

举例:

A 序列:	4	2	6	5	1	3	2
X 序列:	4	-2	4	-1	-4	2	-1

差 分 数 组

性质:

A 序列的区间 +d: O(n) 的操作

X 序列的2点操作: O(1) 的操作

举例:

			+3	+3	+3	+3	
A 序列:	4	2	6	5	1	3	2
X 序列:	4	-2	4	-1	-4	2	-1

差 分 数 组

性质:

A 序列的区间 +d: O(n) 的操作

X 序列的2点操作: O(1) 的操作

举例:

			+3	+3	+3	+3	
A 序列:	4	2	6	5	1	3	2
			+3				**-3**
X 序列:	4	-2	**4**	-1	-4	2	**-1**

A[i~j]+d => X[i]+d, X[j+1]-d

差分数组-例题

1. P1047：校门外的树
2. HZOJ-233：最高的奶牛
3. P1083 [NOIP2012 提高组] 借教室

二、树状数组

1. 树状数组基础知识
2. 树状数组-课后练习题

再 看 前 缀 和

前缀和数组:

<u>**初始化**</u>: O(n) 时间复杂度，顺序扫描原数组即可

<u>**查询区间和**</u>: O(1) 时间复杂度，S[j]-S[i]即为原数组 i 到 j 的区间和

<u>**单点修改**</u>: O(n) 时间复杂度，需要修改 S[i]~S[n] 所有值

慢，是因为 S[i] 的值与之前原数组中所有项都有关系

<u>弱化这种关系</u>，即可加快单点修改速度，当然也会丧失部分查询速度

可这种取舍，是值得的!

lowbit函数

定义:

lowbit(i)：代表 i 这个数字，二进制表示的最后一位 1 的位权

例如:

lowbit(8) = $(1000)_2$ = 8

lowbit(6) = $(110)_2$ = 2

lowbit(12) = $(1100)_2$ = 4

lowbit(7) = $(111)_2$ = 1

lowbit(x) = x & (-x)

树 状 数 组

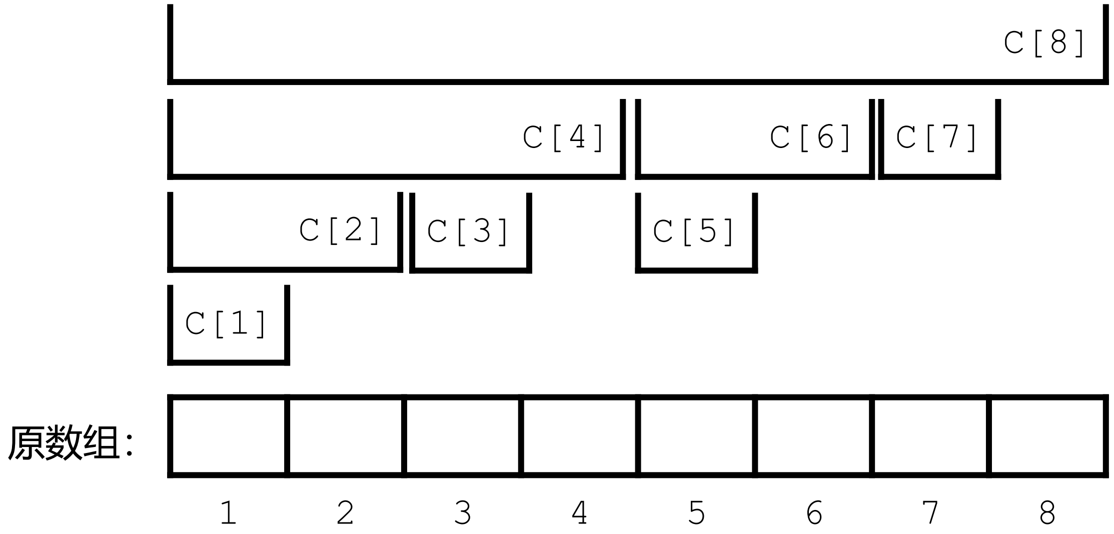
改进前缀和:

lowbit(i): 代表 C[i] 代表前 lowbit(i) 项的和

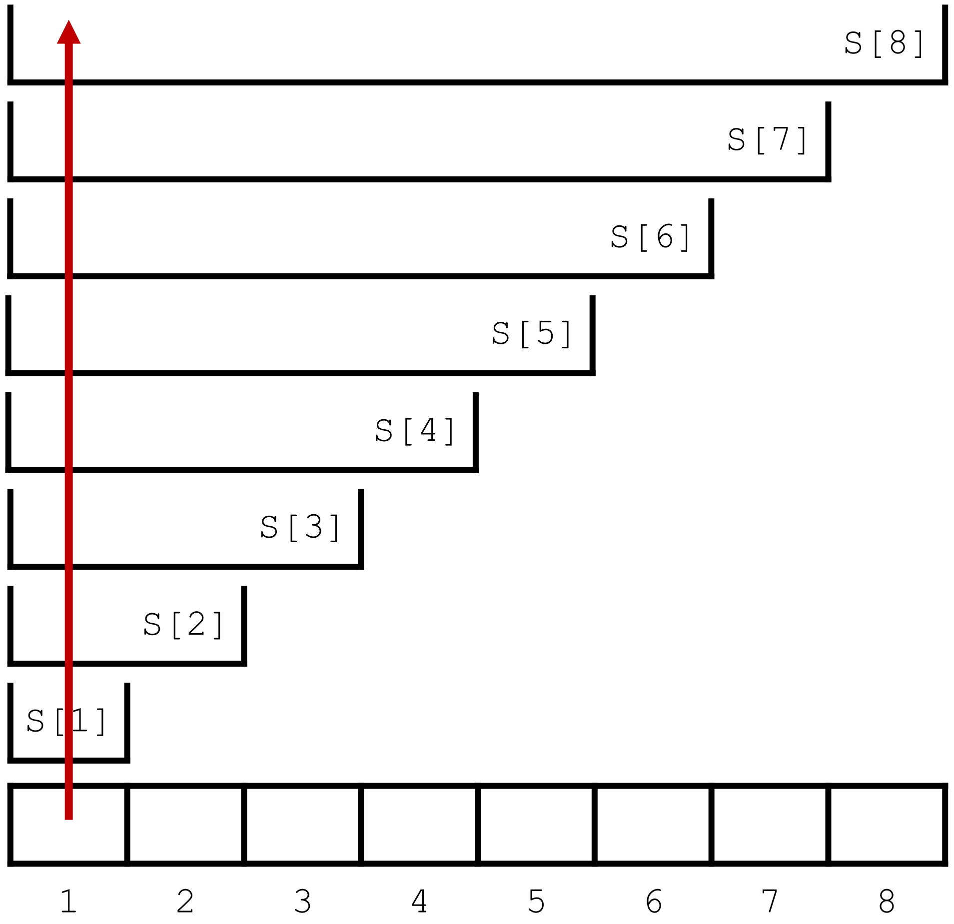
例如:

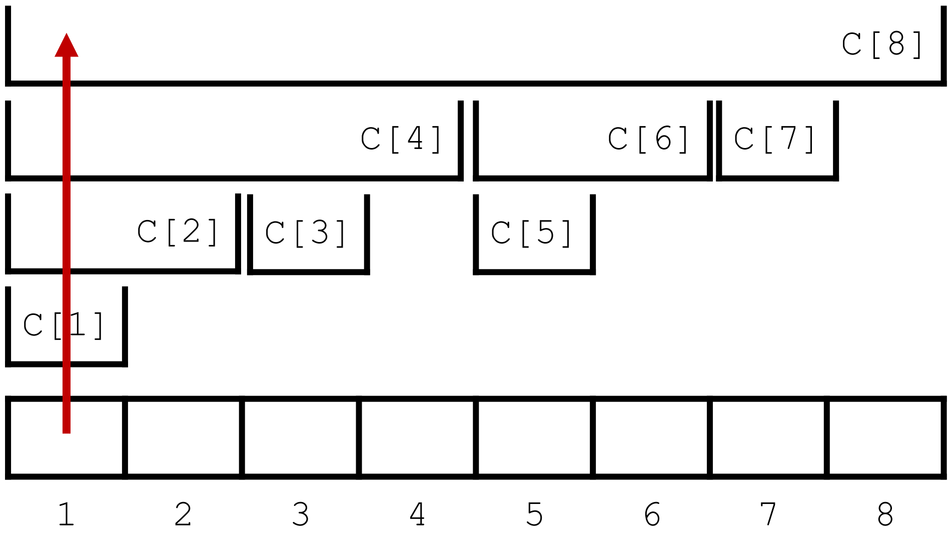
lowbit(10) = 2, C[10] = a[10]+a[9]

lowbit(12) = 4, C[12] = a[12]+a[11]+a[10]+[9]



树 状 数 组





树 状 数 组

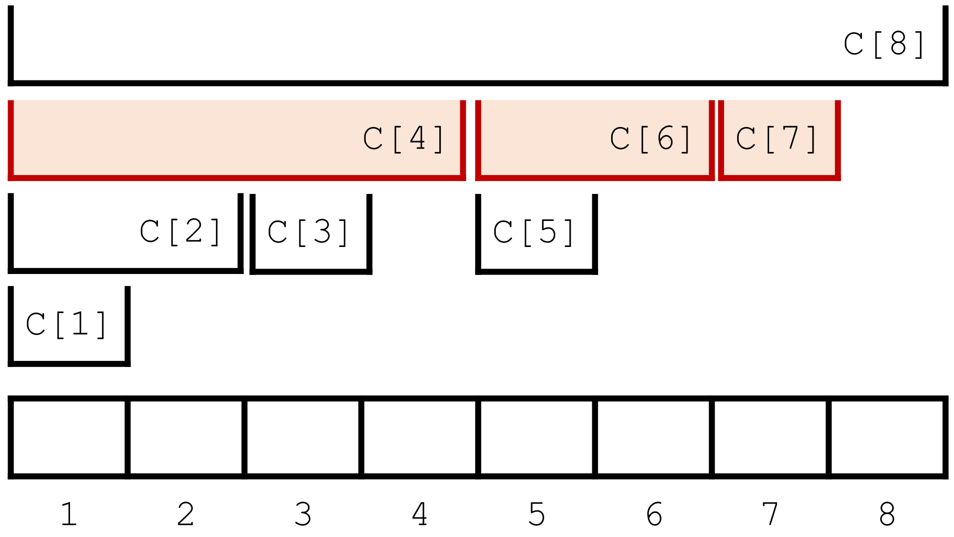
基本操作:

前缀和查询: S[i]=S[i-lowbit(i)]+C[i]

例如:

S[7] = S[6]+C[7] = S[4]+C[6]+C[7] = C[4]+C[6]+C[7]

S[12]=?



树 状 数 组

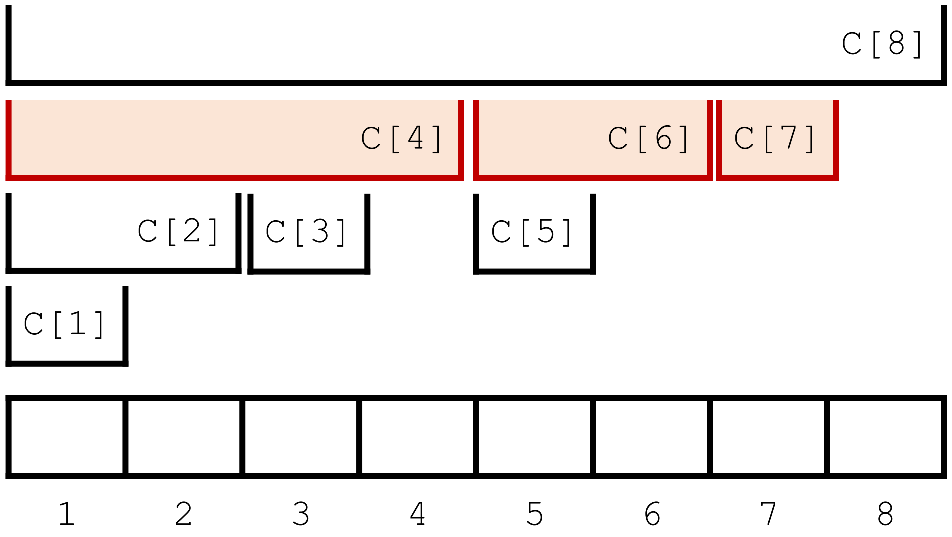
基本操作:

前缀和查询: S[i]=S[i-lowbit(i)]+C[i]

例如:

S[7] = S[6]+C[7] = S[4]+C[6]+C[7] = C[4]+C[6]+C[7]

S[12]= S[8]+C[12] = C[8]+C[12]



树 状 数 组

基本操作:

单点修改：A[j]发生改变时，当修改完C[j]，下一个应该修改 C[j+lowbit[j]]

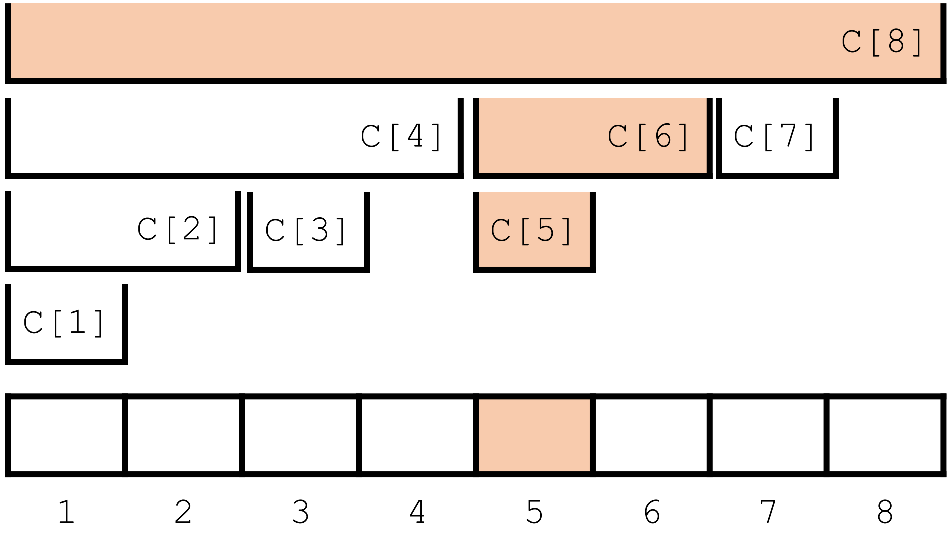
例如:

更新原数组 A[5] 的值，那么需要更新:

C[5], 5 + lowbit(5) = 6,

C[6], 6 + lowbit(6) = 8

C[8] 这三个点的值



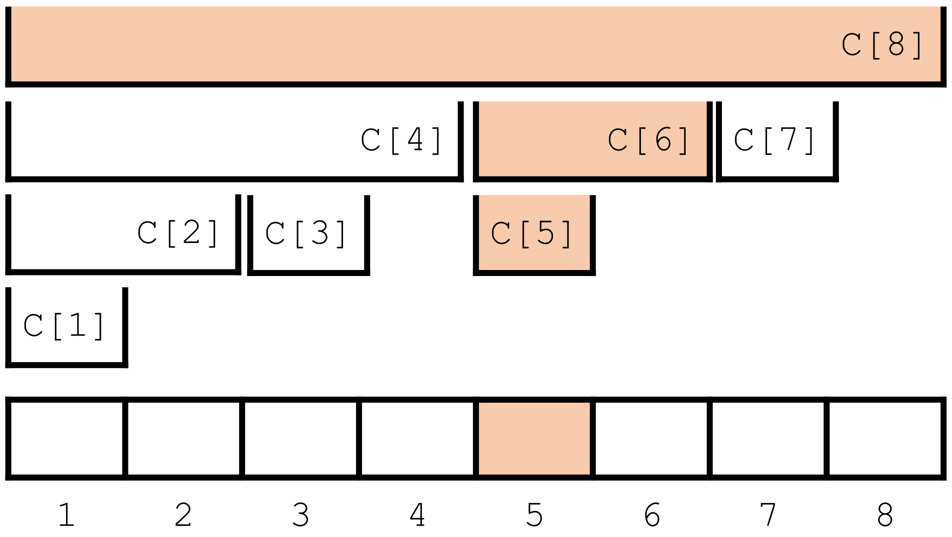
树 状 数 组

基本操作:

单点修改: 当修改A[j]位置的值的时候，首先需要更新的显然是C[j]的值，可C[j]之后，应该更新哪个值呢？也就是找到C[j]脑袋上面的区间。

例如:

更新原数组 A[5] 的值，那么需要更新 C[5],C[6],C[8] 这三个点的值



树 状 数 组

基本操作:

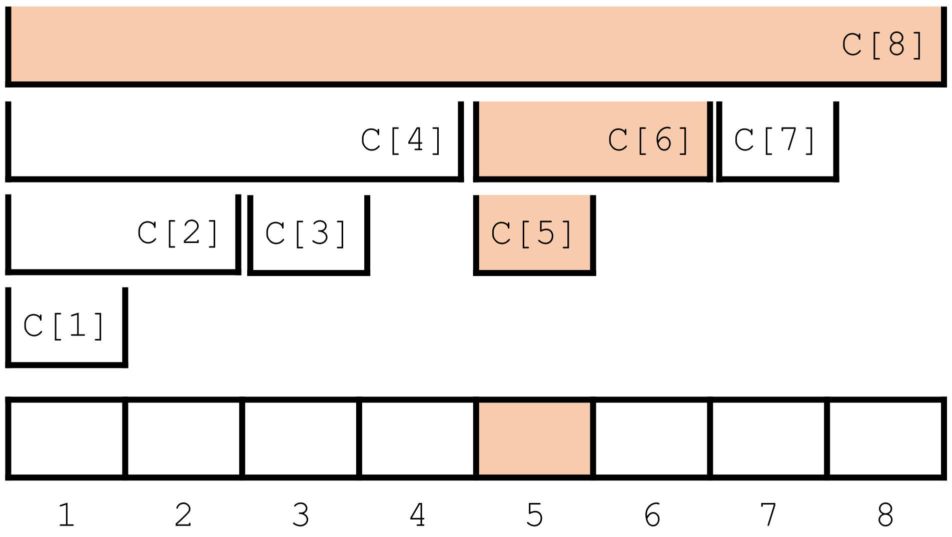
性质1: C[j+k] 当 k < lowbit(j) 时, C[j+k]区间不包含 C[j]区间

证明1:

易得 lowbit(j+k) <= k

j+k-lowbit(j+k) >= j+k-k

j+k-lowbit(j+k) >= j



树 状 数 组

基本操作：

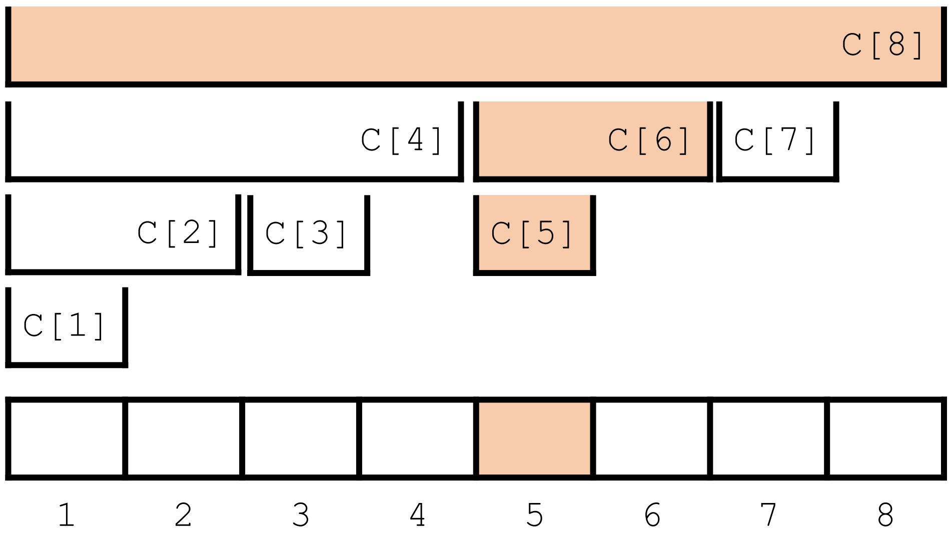
性质2： C[j+k] 当 k = lowbit(j) 时，C[j+k]区间包含 C[j] 区间

证明2：

易得 lowbit(j+k) > k

j+k-lowbit(j+k) < j+k-k

j+k-lowbit(j+k) < j



树状数组—关键词

lowbit函数：求数字 x 中二进制表示的最后一位 1

查 询 操 作：维护前缀和，向前统计，i-lowbit(i)

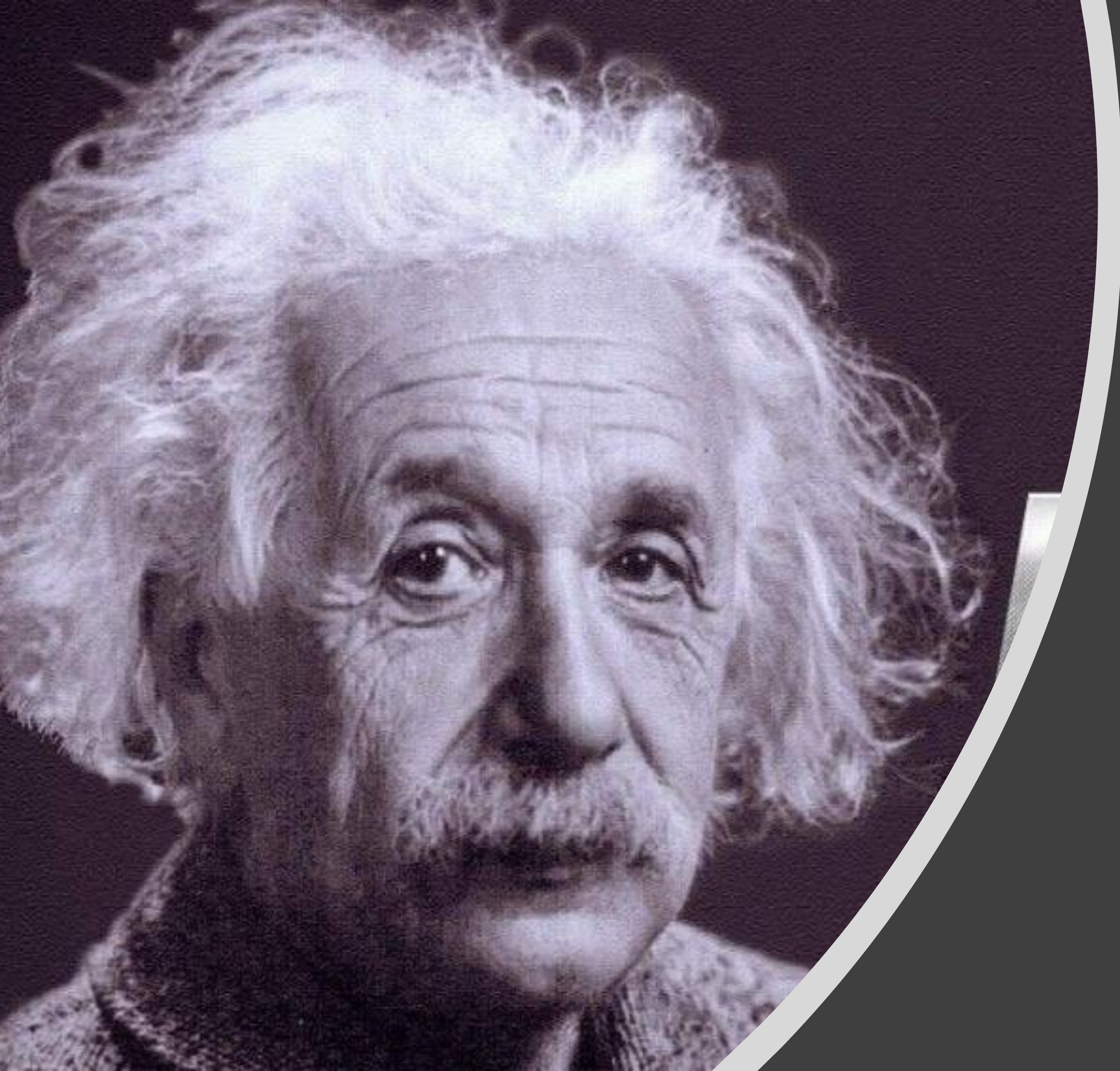
更 新 操 作：更新单点的值，先后更新，i+lowbit(i)

二、树状数组

1. 树状数组基础知识
2. 树状数组-课后练习题

树状数组-例题

1. P3374 【模板】树状数组 1
2. P3368 【模板】树状数组 2



为什么

会出一样的题目？