Algorithm (/tags/#Algorithm)

二叉索引树(树状数组)

Posted by Xsp on June 6, 2017

内容参考 《算法竞赛入门经典-训练指南》

二叉索引树(Binary Indexed Tree, BIT)，俗称树状数组，又以发明者命名为Fenwick树。现多用于高效计算数列的前缀和，区间和。 树状数组-维基百科 (https://zh.wikipedia.org/wiki/%E6%A0%91%E7%8A%B6%E6%95%B0%E7%BB%84)。

应用

给定一个n个元素的数组 A_1 ，$A_2 \ldots A_n$，设计一个查询操作Query(i, j) = $A_i + A_{i+1} + \ldots + A_j$。 如何做？如果用前缀和的思想，计算$S_n = \sum_{i=1}^{n} A_i$ ，那么 Query(i, j) = $S_j - S_i$，单次查询时间为O(1)，但是如果需要更新A_i的话，那每次更新，都需要更新一批S_i，会很慢，所以需要二叉索引树，它解决这个问题的更新及查询的时间复杂度都是O(logn)。

Lowbit函数

定义Lowbit(x) 为 x 的二进制表达式中最右边的1所对应的值。如88的二进制是101 1000， 所以lowbit(88) = 8(二进制是1000)，实现中用lowbit(x) = x & -x。因为计算机中整数采用补码存储，因此 -x 是 x 按位取反，末尾加1以后的结果。两者按位与之后，前面的变为0，就可以得到lowbit。

构造BIT树

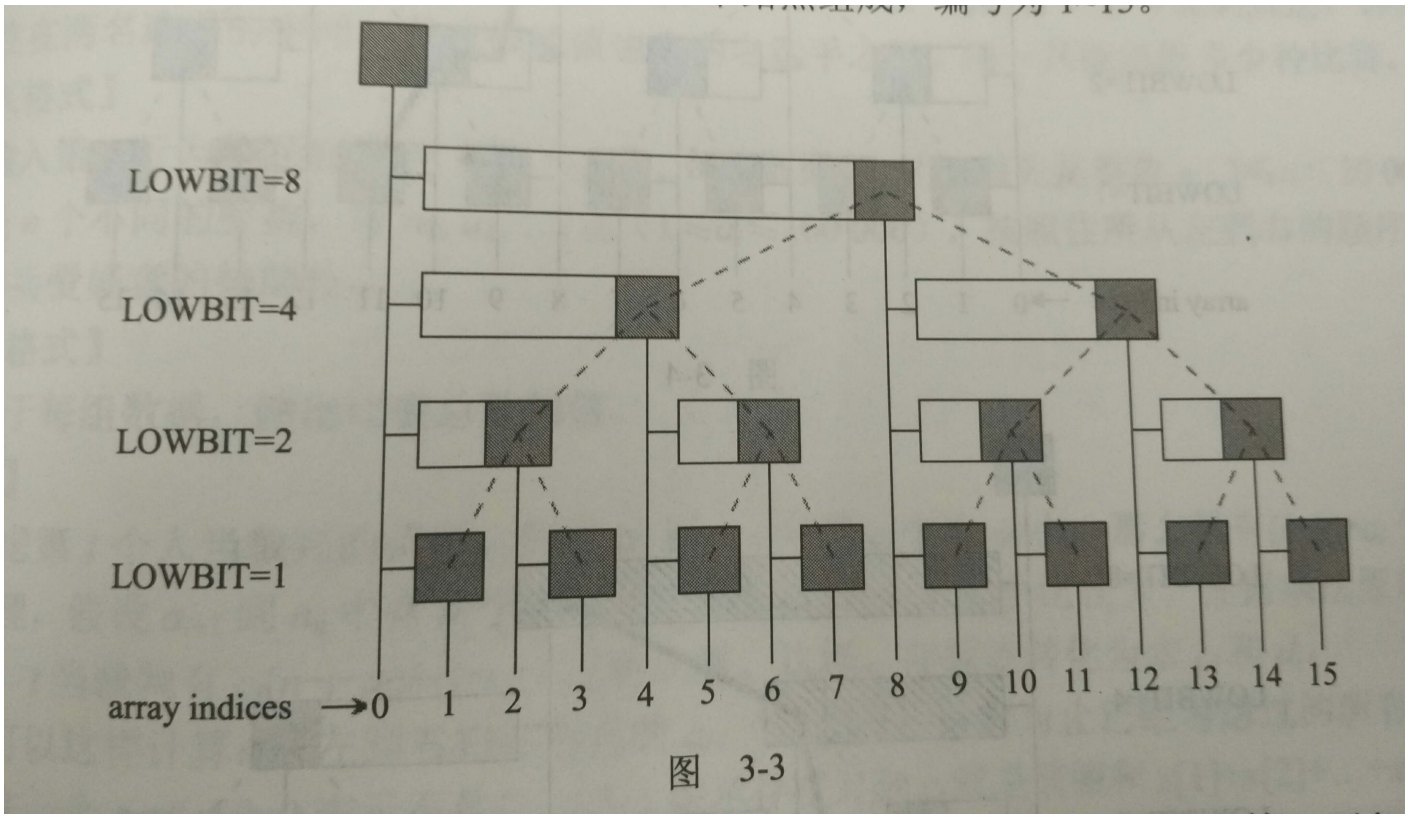

图 3-3

如图3-3所示，对于节点i，如果它是左子结点，那么父节点的编号就是i + lowbit(i)；如果它是右子节点，那么父节点的编号是i - lowbit(i)。需要注意编号为0的点是虚拟节点。之后的定义的数组的序号也是从1 ~ n。

构造一个辅助数组 $C_i = \sum_{j=i-lowbit(i)+1}^{i} A_j$，$C_i$即是A数组中的一段连续和，例如$C_{12} = A_9 + A_{10} + A_{11} + A_{12}$。

求前缀和S_i，在树中从节点i往左上走，把经过的C_i累加。

修改A_i，在树中从节点i往右上走，边走边修改C_i。如下图所示。

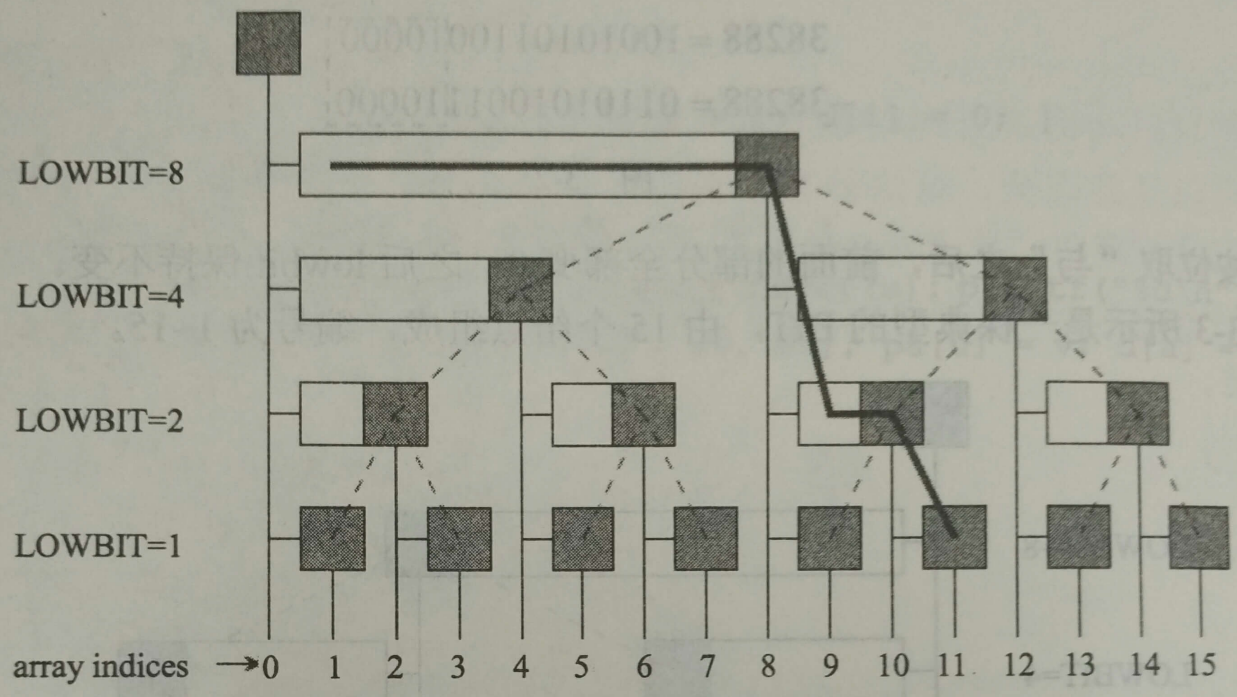

图 3-4

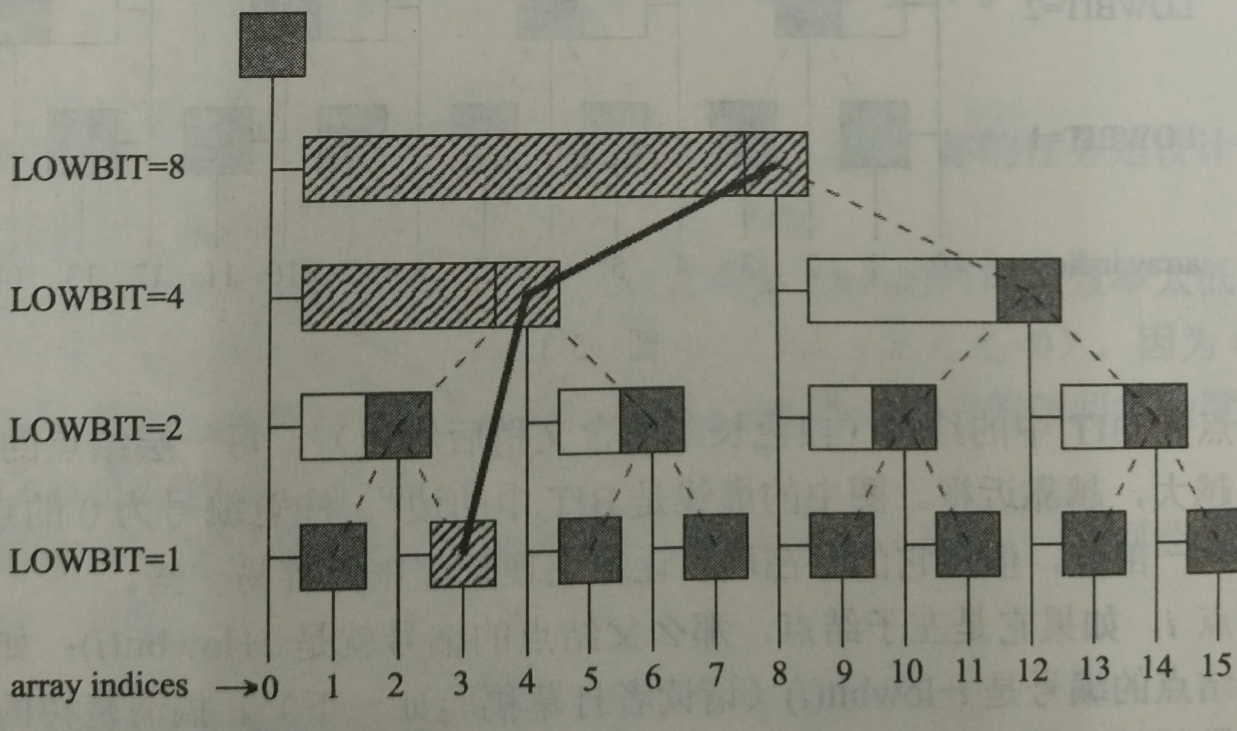

图 3-5

两个操作的代码：

```
// 求和 int sum(int x) {
    int ret = 0;
    while (x > 0) {
        ret += C[x];
        x -= lowbit(x);
    }
    return ret;
}
// 修改，让A[x] 增加d void add(int x, int d) {
    while (x <= n ) {
        C[x] += d;
        x += lowbit(x);
    }
}
```

练习

307. Range Sum Query - Mutable - leetcode (https://leetcode.com/problems/range-sum-query-mutable/#/description)

AC代码 (https://github.com/husterxsp/leetcode/blob/master/307-range-sum-query-mutable/NumArray.cpp)

线段树(区间树)

线段树与二叉索引树结构类似，单次查询和更新的时间复杂度也是$O(logn)$。不过线段树能求解的问题范围更大一些，比如区间和，区间最值，能用二叉索引树解的一般也能用线段树解。待完善。。。

上面那个leetcode 307用线段树始终超时。。。

另外327. Count of Range Sum (https://discuss.leetcode.com/topic/33734/java-segmenttree-solution-36ms) 的线段树解是在没看懂。。

参考：

- http://www.cnblogs.com/xiaoyao24256/p/6590885.html (http://www.cnblogs.com/xiaoyao24256/p/6590885.html)
- http://www.cnblogs.com/TenosDoIt/p/3453089.html (http://www.cnblogs.com/TenosDoIt/p/3453089.html)
- http://blog.csdn.net/disappearedgod/article/details/24425983 (http://blog.csdn.net/disappearedgod/article/details/24425983)
- 线段树-维基百科 (https://zh.wikipedia.org/wiki/%E7%BA%BF%E6%AE%B5%E6%A0%91_(%E5%8C%BA%E9%97%B4%E6%9F%A5%E8%AF%A2))
- 统计的力量 (https://wenku.baidu.com/view/0c1bbba40029bd64783e2cca.html)

PREVIOUS
泛型算法 **(/2017/05/30/GENERIC-ALGORITHM/)**

NEXT
C++位运算 **(/2017/06/07/CPP-BIT-MANIPULATION/)**

FEATURED TAGS **(/tags/)**

前端 (/tags/#前端) 360前端星 (/tags/#360前端星) JavaScript (/tags/#JavaScript) Shell (/tags/#Shell) C++ (/tags/#C++) Algorithm (/tags/#Algorithm)

(https://www.zhihu.com/people/xushaopeng) (http://weibo.com/3824822374)
(https://github.com/husterxsp) (https://www.linkedin.com/in/少鹏-徐-002574132)