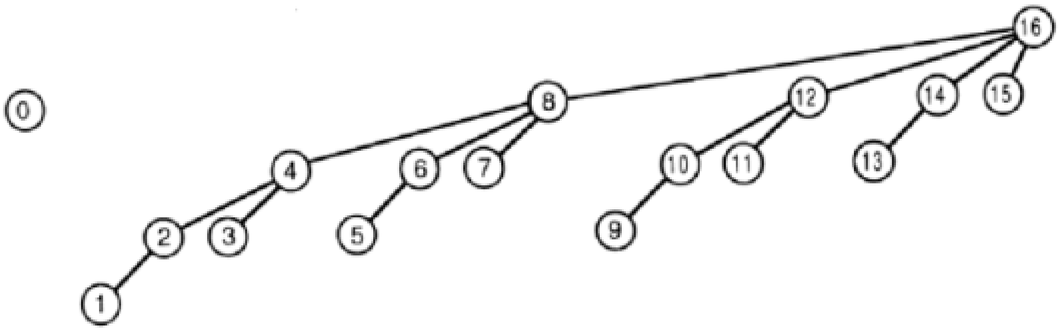
Data Structures and Algorithms with "saanc"

Explore yourself with my ideas :)

ADVANCED DATA STRUCTURES

Binary Indexed Tree (Fenwick Tree)

AUGUST 29, 2014 | SANU KUMAR GUPTA | BINARY INDEXED TREE, CUMULATIVE SUM, DATA STRUCTURES, FENWICK TREE | 13 COMMENTS



(https://sanugupta.files.wordpress.com/2014/08/bit2-1.png)

Introduction:

Binary Indexed Tree (it will be called as **BIT** throughout this post) is an advanced data structure,is often used to store cumulative frequencies and manipulating cumulative frequency table. To understand this better, let’s take an example:

We have n boxes numbered from 1 to n.Then the possible queries are:

1.Put a marble into **i**th box.

2.sum marbles from box l to box r.

Considering the basic approach we can solve both the queries with worst case complexities O(1) and O(n) respectively.But suppose if there are m numbers of query-2 then it will take O(m.n)**.** If we do some effort then we can solve both the queries with worst case complexities O(log n) and O(log n) using **Segment tree**. Same thing can be done using Binary indexed tree with O(logn) and O(logn) complexities. So why one should learn BIT ? There are many reasons for that:

1.Uses less memory than RMQ.

2.Easy to code.

3.Could be used in many problems about number sequence.

The only disadvantage is that it is hard to understand (Don’t worry this statement would be considered as a false statement at the end of this post 😃).

Basic Idea:

We know that we can represent any number as a sum of powers of 2.

example: 22 = 16 + 4 + 2 [2^4 + 2^2 + 2^1]

applying this idea to BIT we are going to express the sum of box[1]…..box[r] as sum of some blocks each of them having 2^k elements. Didn’t get it ? Don’t worry …journey starts from the next line.

Prerequisites:

You need some basic understanding of bitwise operations.Suppose there is a number x.

Think of the statement **x&(-x) .** It will give you the number generated by taking last occurring 1 with all the following 0’s in base 2 representation of the number x.

example: suppose x=20. Its binary representation will be 10**100**. x&(-x) will give you 100 ie 4.

Attention! So basically above example means that sum of box[1]….box[20] can be divided as sum of box[1]….box[16] +sum of box[17]….box[20] (as 20-16=4). So at this instant we want that BIT[16] to store cumulative sum from 1 to 16 and BIT[20] to store cumulative sum from 17 to 20 .

If we take x=22.Then sum of box[1]….box[22] will be divided into box[22]….box[21] +box[20]….box[17]+box[16]….box[1].So here we want that- BIT[22]=box[22]….box[21] BIT[20]=box[20]….box[17] BIT[16]=box[16]….box[1]

To get the positions of the array do this: i=x; i=i- (i&-i);

22 – (22 & (-22)) = 20

20 – (20 & (-20))= 16

16 – (16 & (-16))= 0

See the only image attached in this post and I am sure you visualize the concept.

All things could be done using this simple code(assuming queries and storing of elements in given array are 0-based):

Code to intialize BIT[] :

```
int BIT[100000];
void initializeBIT(int box[],int n)//Array box[] is the given array.
 {
        int i,k;
        memset(BIT,0,sizeof(BIT)); //setting all elements to 0
       for(i=1;i<=n;i++)         //main loop
       {
             int value_to_be_added =box[i-1];
             k=i;
             while(k<=n)
             {
                  BIT[k] +=value_to_be_added;  //adding box[i-1] to a
                  k +=(k & (-k));
             }

      }

 }
```

Code For Query-2:

```
int query(int R,int n)
 {
        int ans=0;
        int index_till =R+1;  //here R+1 is used because query is 0-ba
        while(index_till >0)
        {
               ans +=BIT[index_till];  //Pulling segments to build an
               index_till-=(index_till & (-index_till));//getting the
        }
        return ans;
 }
```

Code for Query-1:

```
 void update(int index,int value,int n)
 {
     int index_to_modify = index+1; //same reason,query is 0-based
     while(index_to_modify <=n)
     {
         BIT[index_to_modify] +=value; //modifying all the necessary
         index_to_modify += (index_to_modify & (-index_to_modify));
     }
 }
```

Please try your hands on it with a simple example.Do it on paper and you will understand what is happening 🙂

Take a visualization of Binary Indexed Tree: Click Here (http://www.comp.nus.edu.sg/~stevenha/visualization/bit.html)

Got the idea? Great 🙂 Now solve the following problems:

Problems:

Inversion count-spoj (http://www.spoj.com/problems/INVCNT/)

Yodaness-spoj (http://www.spoj.com/problems/YODANESS)

Increasing subsequences-spoj (http://www.spoj.com/problems/INCSEQ/)

Horrible queries-spoj (http://www.spoj.com/problems/HORRIBLE/)

Little Girl and Maximum Sum-codeforces (http://www.codeforces.com/problemset/problem/276/C)

Ctrick-spoj (http://www.spoj.com/problems/CTRICK)

References:

Topcoder (http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees), quora (http://www.quora.com/CodeChef/What-are-some-suggestions-for-problems-where-Binary-Indexed-Tree-BIT-Fenwick-Tree-is-the-primary-concept), ahmed-aly (http://www.ahmed-aly.com/Category.jsp?ID=26), gvikei's blog (http://codeforces.com/blog/entry/619),VisuAlgo (http://www.comp.nus.edu.sg/~stevenha/visualization/bit.html). Follow my blog for further posts(click on the right "+follow" symbol 🙂

13 thoughts on "Binary Indexed Tree (Fenwick Tree)"

1. **P_ZHONG** *says:*
   Thank for your sharing.
   Binary indexed tree is amazing!!!!!!!!!!!

   MARCH 6, 2016 AT 12:54 PM | REPLY
2. **ANONYMOUS** *says:*
   Thanks for the clear examples to explain the concept! Explanations on other pages left me more confused than before!

   JANUARY 24, 2016 AT 9:11 AM | REPLY
3. **ANONYMOUS** *says:*
   can you explain me how to solve ctrick using binary tree?

   OCTOBER 11, 2015 AT 7:57 PM | REPLY
4. Pingback: A List Of Some Algorithms with a lot of Resources |

5. Pingback: Getting started with competitive coding | Sameer Chaudhari

6. **ANONYMOUS** *says:*
   raton kumar from Bangladesh
   pls write about "segment tree"
   thanks in advance

   JUNE 20, 2015 AT 12:45 PM | REPLY
7. **ANONYMOUS** *says:*
   well explained. after browsing through lots of youtube videos and articles finally i get it. thank you very much.
   pls clarify the update process…

   array=[1,2,3,4]
   sumarray(0,3) = 10
   update(1,2) // update 2 with value 2

   result should be 10
   but the result is sumaray(0,3) = 12

   JUNE 20, 2015 AT 12:39 PM | REPLY

   - **SANU KUMAR GUPTA** *says:*
     Here update(i,v) represents: value at index i should be increased with value v.
     As much I understood you are assuming that update(i,v) is setting the value at index i to v which is wrong. You can modify this function to do what you want by passing the value v= new value- current value at index i.
     Hope this helps!

     JUNE 20, 2015 AT 6:44 PM | REPLY

8. **ANONYMOUS** *says:*
   Should it not be k -=(k & (-k)); instead of k +=(k & (-k)); in Code to Intialize.

   FEBRUARY 12, 2015 AT 2:15 AM | REPLY

   - **SANU KUMAR GUPTA** *says:*
     No, because for every index …contribution should be added to the next ones

     FEBRUARY 13, 2015 AT 12:35 PM | REPLY

9. **ANONYMOUS** *says:*
   thanks for explaining it in better way 🙂

   DECEMBER 30, 2014 AT 4:36 PM | REPLY

10. Pingback: Data Structures and Algorithms | sbit

11. **ROOKIECODER594** *says:*
    nice explanation

    SEPTEMBER 23, 2014 AT 9:26 PM | REPLY

Blog at WordPress.com.