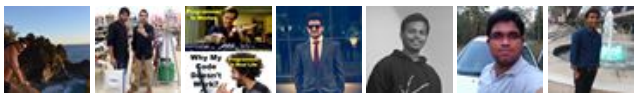
Binary Indexed Tree or Fenwick tree

Let us consider the following problem to understand Binary Indexed Tree.

We have an array arr[0 . . . n-1]. We should be able to
1 Find the sum of first i elements.
2 Change value of a specified element of the array arr[i] = x where 0 <= i <= n-1.

A **simple solution** is to run a loop from 0 to i-1 and calculate sum of elements. To update a value, simply do arr[i] = x. The first operation takes O(n) time and second operation takes O(1) time. Another simple solution is to create another array and store sum from start to i at the i'th index in this array. Sum of a given range can now be calculated in O(1) time, but update operation takes O(n) time now. This works well if the number of query operations are large and very few updates.

Can we perform both the operations in O(log n) time once given the array?



One Efficient Solution is to use Segment Tree that does both operations in O(Logn) time.

Using Binary Indexed Tree, we can do both tasks in O(Logn) time. The advantages of Binary Indexed Tree over Segment are, requires less space and very easy to implement..

Representation

Binary Indexed Tree is represented as an array. Let the array be BITree[]. Each node of Binary Indexed Tree stores sum of some elements of given array. Size of Binary Indexed Tree is equal to n where n is size of input array. In the below code, we have used size as n+1 for ease of implementation.

Construction

We construct the Binary Indexed Tree by first initializing all values in BITree[] as 0. Then we call update() operation for all indexes to store actual sums, update is discussed below.

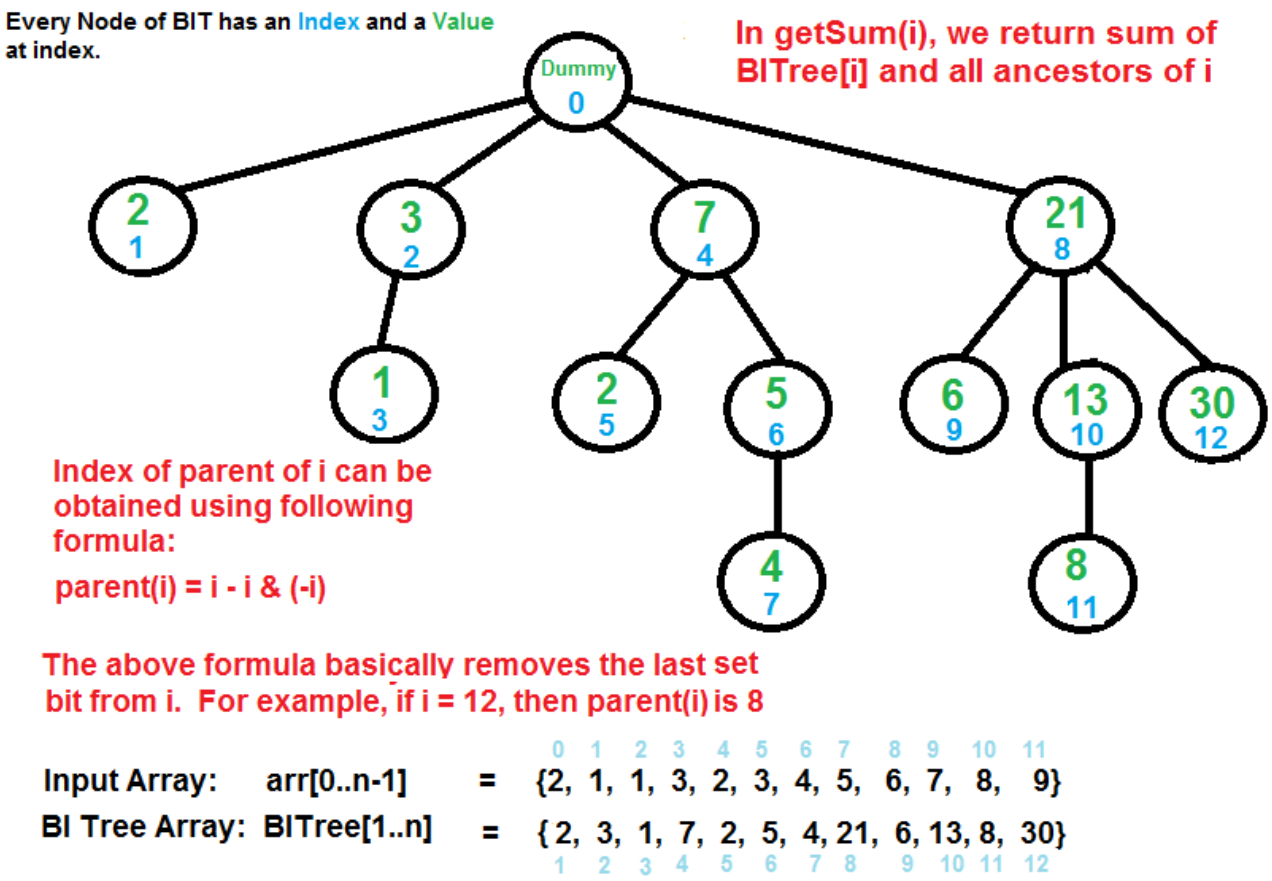
Operations

```
getSum(index): Returns sum of arr[0..index]
// Returns sum of arr[0..index] using BITree[0..n].  It assumes t
hat
// BITree[] is constructed for given array arr[0..n-1]
1) Initialize sum as 0 and index as index+1.
2) Do following while index is greater than 0.
...a) Add BITree[index] to sum
...b) Go to parent of BITree[index].  Parent can be obtained by r
emoving
```


Popular Posts

- Top 10 Algorithms and Data Structures for Competitive

```
     the last set bit from index, i.e., index = index - (index &
(-index))
3) Return sum.
```



View of Binary Indexed Tree to unserstand getSum() operation

The above diagram demonstrates working of getSum(). Following are some important observations.

Node at index 0 is a dummy node.

A node at index y is parent of a node at index x, iff y can be obtained by removing

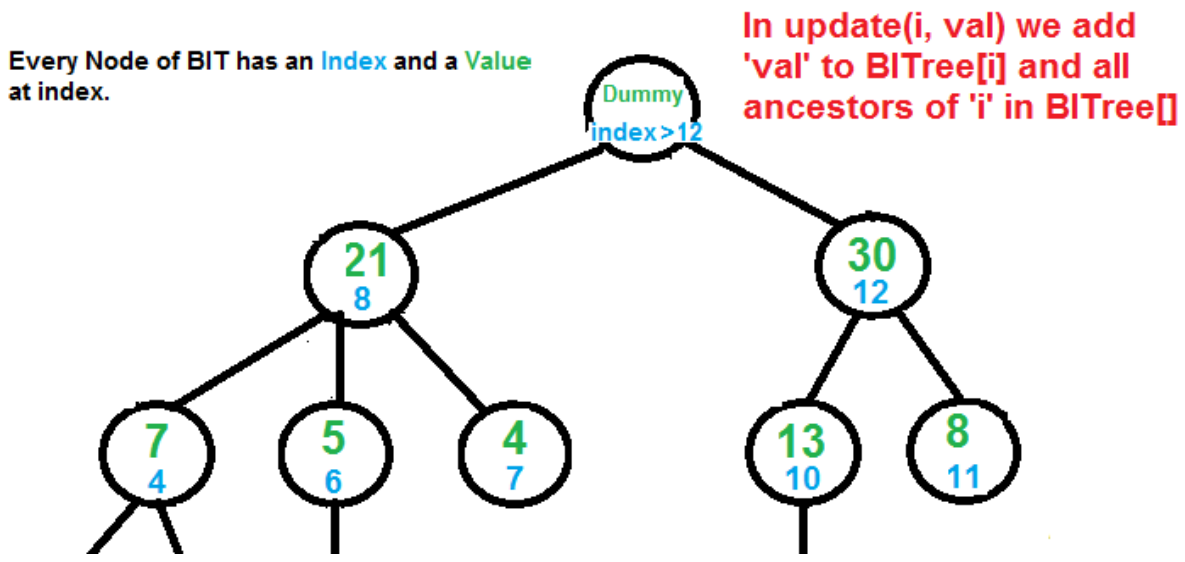
Programming

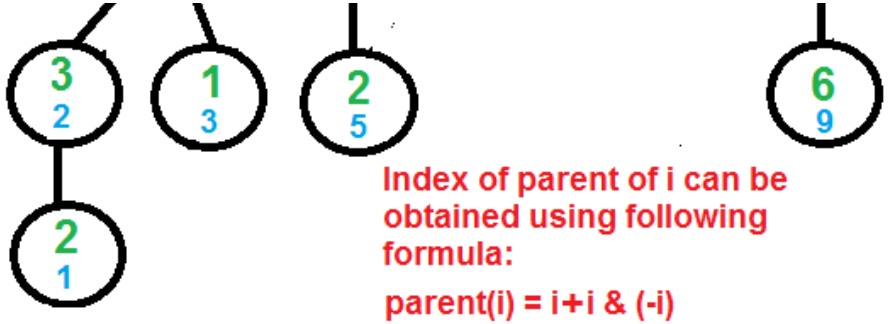
- Top 10 algorithms in Interview Questions
- How to begin with Competitive Programming?
- All permutations of a given string
- Memory Layout of C Programs
- Understanding “extern” keyword in C
- Heavy Light Decomposition
- Sorted Linked List to Balanced BST
- Comb Sort
- Aho-Corasick Algorithm for Pattern Searching

last set bit from binary representation of x.

A child x of a node y stores sum of elements from of y(exclusive y) and of x(inclusive x).

```
update(index, val): Updates BIT for operation arr[index] += val
// Note that arr[] is not changed here.  It changes
// only BI Tree for the already made change in arr[].
1) Initialize index as index+1.
2) Do following while index is smaller than or equal to n.
...a) Add value to BITree[index]
...b) Go to parent of BITree[index].  Parent can be obtained by r
emoving
     the last set bit from index, i.e., index = index + (index &
(-index))
```





Contents of arr[] and BITree[] are same as above diagram for getSum()

View of Binary Indexed Tree to understand update() operation

The update process needs to make sure that all BITree nodes that have arr[i] as part of the section they cover must be updated. We get all such nodes of BITree by repeatedly adding the decimal number corresponding to the last set bit.

How does Binary Indexed Tree work?

The idea is based on the fact that all positive integers can be represented as sum of powers of 2. For example 19 can be represented as 16 + 2 + 1. Every node of BI Tree stores sum of n elements where n is a power of 2. For example, in the above first diagram for getSum(), sum of first 12 elements can be obtained by sum of last 4 elements (from 9 to 12) plus sum of 8 elements (from 1 to 8). The number of set bits in binary representation of a number n is O(Logn). Therefore, we traverse at-most O(Logn) nodes in both getSum() and update() operations. Time complexity of construction is O(nLogn) as it calls update() for all n elements.

- Recursion

All Categories

Select Category

Recent Comments

Rashmi Mishra check this solution...

Check if two given strings are at edit distance one · 1 hour ago

Rashmi Mishra sorry my bad

Reverse an array without affecting special characters · 1 hour ago

Rashmi Mishra I doubt that this question can be solved...

Check if two given strings are at edit distance one · 1 hour ago

Ambuj Mani Tripathi GeeksforGeeks what is the need of 4th Argument...

Find a peak element · 1 hour ago

Pratyush Singh the best book for beginner to learning C

C · 1 hour ago

Implementation:

Following is C++ implementation of Binary Indexed Tree.

```
// C++ code to demonstrate operations of Binary Index Tree
#include <iostream>
using namespace std;

/*            n  --> No. of elements present in input array.
    BITree[0..n] --> Array that represents Binary Indexed Tree.
    arr[0..n-1]  --> Input array for whic prefix sum is evaluated
. */

// Returns sum of arr[0..index]. This function assumes
// that the array is preprocessed and partial sums of
// array elements are stored in BITree[].
int getSum(int BITree[], int index)
{
    int sum = 0; // Iniialize result

    // index in BITree[] is 1 more than the index in arr[]
    index = index + 1;

    // Traverse ancestors of BITree[index]
    while (index>0)
    {
        // Add current element of BITree to sum
        sum += BITree[index];

        // Move index to parent node in getSum View
        index -= index & (-index);
    }
```

Aditya Siddheshwar Upadhyay if i have given a string as an input. then how...

Rearrange a string so that all same characters become at least d distance away · 2 hours ago

Tags

Adobe Advance Data Structures Advanced Data Structures Amazon array Backtracking Bharti SoftBank (HIKE) Bit Magic C++ CN c puzzle D-E-Shaw DBMS Divide and Conquer Dynamic Programming Flipkart GATE GATE-CS-2012 GATE-CS-C-Language GATE-CS-DS-&-Algo GATE-CS-Older GFacts Goldman Sachs Google Graph Greedy Algorithm Hashing Interview Experience Java MAQ Software MathematicalAlgo Matrix Microsoft Morgan Stanley Operating systems

```
    return sum;
}

// Updates a node in Binary Index Tree (BITree) at given index
// in BITree.  The given value 'val' is added to BITree[i] and
// all of its ancestors in tree.
void updateBIT(int BITree[], int n, int index, int val)
{
    // index in BITree[] is 1 more than the index in arr[]
    index = index + 1;

    // Traverse all ancestors and add 'val'
    while (index <= n)
    {
       // Add 'val' to current node of BI Tree
       BITree[index] += val;

       // Update index to that of parent in update View
       index += index & (-index);
    }
}

// Constructs and returns a Binary Indexed Tree for given
// array of size n.
int *constructBITree(int arr[], int n)
{
    // Create and initialize BITree[] as 0
    int *BITree = new int[n+1];
    for (int i=1; i<=n; i++)
        BITree[i] = 0;
```

```
    // Store the actual values in BITree[] using update()
    for (int i=0; i<n; i++)
        updateBIT(BITree, n, i, arr[i]);

    // Uncomment below lines to see contents of BITree[]
    //for (int i=1; i<=n; i++)
    //      cout << BITree[i] << " ";

    return BITree;
}


// Driver program to test above functions
int main()
{
    int freq[] = {2, 1, 1, 3, 2, 3, 4, 5, 6, 7, 8, 9};
    int n = sizeof(freq)/sizeof(freq[0]);
    int *BITree = constructBITree(freq, n);
    cout << "Sum of elements in arr[0..5] is "
         << getSum(BITree, 5);

    // Let use test the update operation
    freq[3] += 6;
    updateBIT(BITree, n, 3, 6); //Update BIT for above change in
arr[]

    cout << "\nSum of elements in arr[0..5] after update is "
         << getSum(BITree, 5);

    return 0;
}
```

Output:

```
Sum of elements in arr[0..5] is 12
Sum of elements in arr[0..5] after update is 18
```

Can we extend the Binary Indexed Tree for range Sum in Logn time?

This is simple to answer. The rangeSum(l, r) can be obtained as getSum(r) – getSum(l-1).

Applications:

Used to implement the arithmetic coding algorithm. Development of operations it supports were primarily motivated by use in that case. See this for more details.

References:

http://en.wikipedia.org/wiki/Fenwick_tree

http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above

54 Comments Category: Advanced Data Structure Tags: Advance Data Structures, Advanced Data Structures

Related Posts:

- Centroid Decomposition of Tree
- Gomory-Hu Tree | Set 1 (Introduction)
- kasai's Algorithm for Construction of LCP array from Suffix Array
- Overview of Data Structures | Set 3 (Graph, Trie, Segment Tree and Suffix Tree)
- Heavy Light Decomposition | Set 2 (Implementation)
- Heavy Light Decomposition | Set 1 (Introduction)
- Count Inversions of size three in a give array

- Count inversions in an array | Set 3 (Using BIT)

Previous post in category Next post in category

54 Comments GeeksforGeeks

xxmajia • 22 days ago

in update step, 2) -> b) as below, the explanation is incorrect, but only diff is + and - in the formula

'Go to parent of BITree[index]. Parent can be obtained by removing

the last set bit from index, i.e., index = index + (index & (-index))'

I recommend this vedio: https://www.youtube.com/watch?...
and I think the right method name for the update step is: getNext

  • Reply • Share ›

Raj • a month ago

https://www.hackerearth.com/no... contains more detailed explana
behind the DS. The parents in sum and update operations of a no
the nodes who are path of the traversal from the root to a particula

  • Reply • Share ›

ankush • 2 months ago

in first diagram, the parent 8 has three children.. Is it still a binary t

  • Reply • Share ›

Nikita Tovstoles ↗ ankush • a month ago

No, it is a tree indexed by binary numbers (0,1). Hence bin

  • Reply • Share ›

Shubham Marathia ↗ ankush • 2 months ago

Actually its named binary tree as we use binary representa
do with binary tree.

  • Reply • Share ›

warrier • 2 months ago

can i get the sum between 3rd and 6th element or some other sur

tree

Reply • Share ›

Nikita Tovstoles warrier • a month ago

yes : subtract smaller-range prefix sum from the larger - ie to include the lower boundary

Reply • Share ›

warrier • 2 months ago

can we use fenwick tree for a range to get sum

Reply • Share ›

jack2684 • 2 months ago

A comment in update(index, val) step 2.(b) I feel quite misleading. succeeding siblings of my root parent. I guess this mistake was m get(index) step 2.(b)

Quote:
"...b) Go to parent of BITree[index]. Parent can be obtained by rem the last set bit from index, i.e., index = index + (index & (-index))"

1 Reply • Share ›

aranjan2012 • 3 months ago

I am a little confused because the figure is not a binary tree. Is tha

Reply • Share ›

oshev aranjan2012 • 2 months ago

children, not to the known type of the tree.

Reply • Share ›

Esha Palta Tripathi • 3 months ago

parent(i) = i - (i & (-i)) is incorrect. It should be :
parent(i) = i - (i & (i - 1))

Reply • Share ›

BARUN HALDER Esha Palta Tripathi • 2 months ago

both are giving same result, check once.

Reply • Share ›

iamlion • 3 months ago

IF i update the element with 0 it does not return correct answer !!!

Reply • Share ›

Amit Kumar Pradhan • 3 months ago

Thank you for the nice article

Reply • Share ›

Dawei Li • 3 months ago

What should be the correct description for the pseudo code of up
It should not be the same as the getSum 2) b). At least the implen

Can some one exaplain???

Reply • Share ›

GeeksforGeeks Mod ➦ Dawei Li • 3 months ago

Thanks for pointing this out. We have updated the pseudo

  • Reply • Share ›

Aleksandar Doykov • 4 months ago

The link to top coder is no longer active. use https://www.topcode

  • Reply • Share ›

RAVI RAJAK ➦ Aleksandar Doykov • 4 months ago

thanks for the link

  • Reply • Share ›

RAVI RAJAK • 4 months ago

```
package ravi.tree.bit;

public class CumulativeFrequency {

protected int [] biTree;

CumulativeFrequency(int n){

biTree = new int[n + 1];

initialize();

}

protected void initialize(){
```

for(int i = 0 ; i < biTree.length ; ++i){

biTree[i] = 0;

}

see more

• Reply • Share ›

Yunfeng Xi ➦ RAVI RAJAK • 3 months ago

What if I want to UPDATE(int i, int j) and QUERY(int i)? Ca

• Reply • Share ›

RAVI RAJAK ➦ Yunfeng Xi • 3 months ago

The structure will need a slight modification with up
index and all its fenwick parent nodes in log n times

• Reply • Share ›

Yunfeng Xi ➦ RAVI RAJAK • 3 months ago

https://www.topcoder.com/commu...

this is an example problem in this article, bu

• Reply • Share ›

NP • 6 months ago

@GeeksforGeeks

I think, 2)..b) step in pseudo code mentioned for update BIT opera

"2)..b) Go to parent of BITree[index]. Parent can be obtained by re

the last set bit from index, i.e., index = index - (index & (-index))"

should be.... index = index + (index & (-index))

Reply • Share ›

GeeksforGeeks Mod NP • 3 months ago

Thanks for pointing this out. We have updated the pseudo

Reply • Share ›

PrateekSingh Chauhan • 6 months ago

in update operation first we will calculate diff between old and new
please correct me if i am wrong

Reply • Share ›

Billionaire • 7 months ago

I prefer segment tree over BIT, and I think segment tree is easier t

1 Reply • Share ›

Vivek Chicharito Zakwalia • 7 months ago

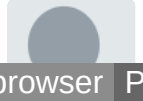
wonder how people get ideas like that! _/_

Reply • Share ›

Sagar • 8 months ago

@GeeksforGeeks THere is pointer missing in the getsum functi

Reply • Share ›



Aman • 8 months ago

Do look into the following explanation also:
http://cs.stackexchange.com/qu...

Reply • Share ›

Name • 8 months ago
Awesome post! Intrigued by the spectacular inner working of Fenv

Reply • Share ›

Asmita Metrewar • 9 months ago
BIT is better than segment tree in case of range update queries. It segment tree takes O(n) time for the same. That has to be the ke
Ref: http://programmingcontests.quo...

Reply • Share ›

Abhijeet Dubey Asmita Metrewar • 8 months ago
For range updates you can use Lazy Propagation with seg for update operation.
Ref : http://se7so.blogspot.in/2012/...

Reply • Share ›

Mission Peace • a year ago
Watch this video for fenwick tree explanation
https://youtu.be/CWDQJGaN1gY

4 Reply • Share ›

AB_kyusak Mission Peace • 6 months ago
nice video

• Reply • Share ›

Koustav Chatterjee Mission Peace • 8 months ago

Khub bhalo video.

• Reply • Share ›

Satty Mission Peace • 9 months ago

indeed a gr8 tutorial, nicely explained.!!

• Reply • Share ›

VivekH Mission Peace • 10 months ago

This is the best tutorial I have found on fenwick tree so far.

• Reply • Share ›

Rajat Narang • a year ago

Great Tutorial, simpler than topcoder's tutorial!!

• Reply • Share ›

Robin Rajat Narang • a year ago

I completely agree :)

• Reply • Share ›

Phi • a year ago

Type in the article with respect to update operation. You update th
is, i + i & -i sends you down the tree, not up the tree, like i - i & -i w

• Reply • Share ›

GeeksforGeeks Mod ➚ Phi • a year ago

Please take a closer look. There are two views of BIT, one

▲ | ▼ • Reply • Share ›

Phi ➚ Phi • a year ago

typo*

▲ | ▼ • Reply • Share ›

manu • a year ago

i think we can do this simply using concept of cumulative frequenc
it will take constant time for both operations

▲ | ▼ • Reply • Share ›

Bhavesh Munot ➚ manu • a year ago

Well, that would have worked if array was not updated at r
execution. So It wont work because, to have O(1) for getsu
update() query.

▲ | ▼ • Reply • Share ›

Siddhanjay Godre • a year ago

I think rangeSum(l,r) to find the sum in range [l,r] should be getsur

Also it is worth mentioning that a fenwick tree can also be used fo
1) Range Update and Point Queries

updating the range [l,r] with the value 'val' ,
update(l,val);

2) Range Update and Range Queries
Construct two Fenwick trees B1 and B2
Now Update can be done as

update(a, b, v)

{
update(B1, a, v)
update(B1, b + 1, -v)
update(B2, a, v * (a-1))

see more

2 ^ | v • Reply • Share ›

kumar gaurav Siddhanjay Godre • a year ago
Could you please explain the 'Range Update and Range Q
^ | v • Reply • Share ›

Siddhanjay Godre kumar gaurav • a year ago
Consider the case of single update of adding 'val' ir
3 cases are possible:
1. 0 ≤ x < a =>0
2. a ≤ x ≤ b =>val * (x - (a-1))
3. b < x < n => val * (b - (a-1))

Now if we can find (val*x), we can get query(0,x) by

2. a ≤ x ≤ b: =>Sum = val*x-val*(a-1) So T = val*(a
3. b < x < n : Sum = 0 So,T = -val*b + val*(a-1)

For maintaining this extra factor of 'T' we can use a

For updating val in [a…b] we :
update(B1,a,val) and update(B1,b,-val) in first BIT a
update(B2, a, val * (a-1)) and Update(B2, b+1, -val

For query(0,x):
query(B1,x)*x - query(B2,x)

see

2 ^ | v • Reply • Share ›

GeeksforGeeks Mod ➦ Siddhanjay Godre • a year ago

Thanks for pointing this out. We have updated formula for well.

1 ^ | v • Reply • Share ›

Guest • a year ago

http://kartikkukreja.wordpress...

2 ^ | v • Reply • Share ›

Load more comments