Segment Tree | Set 1 (Sum of given range)

Let us consider the following problem to understand Segment Trees.

We have an array arr[0 . . . n-1]. We should be able to
1 Find the sum of elements from index l to r where 0 <= l <= r <= n-1
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 l to r and calculate sum of elements in given range. To update a value, simply do arr[i] = x. The first operation takes O(n) time and second operation takes O(1) time.

Another solution is to create another array and store sum from start to i at the ith 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.

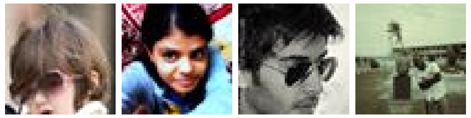
What if the number of query and updates are equal? **Can we perform both the operations in O(log n) time once given the array?** We can use a Segment Tree to do both operations in O(Logn) time.

Representation of Segment trees
1. Leaf Nodes are the elements of the input array.
2. Each internal node represents some merging of the leaf nodes. The merging may be different for different problems. For this problem, merging is sum of leaves under a node.

An array representation of tree is used to represent Segment Trees. For each node at index i, the left child is at index 2*i+1, right child at 2*i+2 and the parent is at $\lfloor (i-1)/2 \rfloor$.

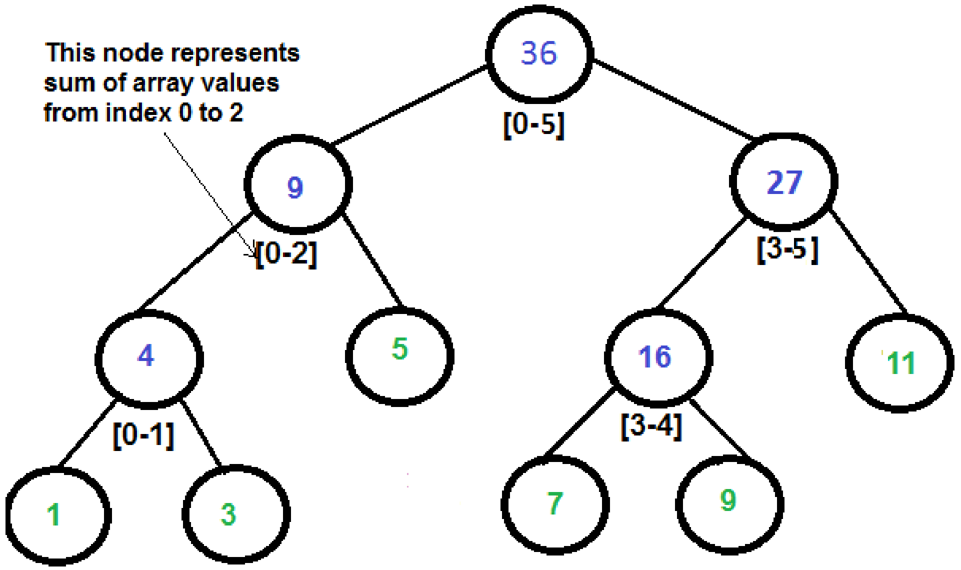
Google™ Custom Search

GeeksforGeeks
86,550 people like GeeksforGeeks.



ITT Tech - Official Site
Tech-Oriented Degree Programs. Education for the Future.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com



Segment Tree for input array {1, 3, 5, 7, 9, 11}

Construction of Segment Tree from given array
We start with a segment arr[0 . . . n-1]. and every time we divide the current segment into two halves(if it has not yet become a segment of length 1), and then call the same procedure on both halves, and for each such segment we store the sum in corresponding node.
All levels of the constructed segment tree will be completely filled except the last level. Also, the tree will be a Full Binary Tree because we always divide segments in two halves at every level. Since the constructed tree is always full binary tree with n leaves, there will be n-1 internal nodes. So total number of nodes will be 2*n – 1.
Height of the segment tree will be $\lceil \log_2 n \rceil$. Since the tree is represented using array and relation between parent and child indexes must be maintained, size of memory allocated for segment tree will be $2 * 2^{\lceil \log_2 n \rceil} - 1$.

Query for Sum of given range
Once the tree is constructed, how to get the sum using the constructed segment tree. Following is algorithm to get the sum of elements.

```
int getSum(node, l, r)
{
   if range of node is within l and r
```

Interview Experiences
Advanced Data Structures
Dynamic Programming
Greedy Algorithms
Backtracking
Pattern Searching
Divide & Conquer
Mathematical Algorithms
Recursion
Geometric Algorithms

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
        return value in node
   else if range of node is completely outside l and r
        return 0
   else
    return getSum(node's left child, l, r) +
           getSum(node's right child, l, r)
}
```

Update a value
Like tree construction and query operations, update can also be done recursively. We are given an index which needs to updated. Let *diff* be the value to be added. We start from root of the segment tree, and add *diff* to all nodes which have given index in their range. If a node doesn't have given index in its range, we don't make any changes to that node.

Implementation:
Following is implementation of segment tree. The program implements construction of segment tree for any given array. It also implements query and update operations.

```
// Program to show segment tree operations like construction, query an
#include <stdio.h>
#include <math.h>

// A utility function to get the middle index from corner indexes.
int getMid(int s, int e) {  return s + (e -s)/2;  }

/*  A recursive function to get the sum of values in given range of th
    The following are parameters for this function.

    st    --> Pointer to segment tree
    index --> Index of current node in the segment tree. Initially 0 i
              passed as root is always at index 0
    ss & se  --> Starting and ending indexes of the segment represente
                 current node, i.e., st[index]
    qs & qe  --> Starting and ending indexes of query range */
int getSumUtil(int *st, int ss, int se, int qs, int qe, int index)
{
    // If segment of this node is a part of given range, then return t
    // sum of the segment
    if (qs <= ss && qe >= se)
        return st[index];

    // If segment of this node is outside the given range
    if (se < qs || ss > qe)
        return 0;
```

Popular Posts

All permutations of a given string
Memory Layout of C Programs
Understanding “extern” keyword in C
Median of two sorted arrays
Tree traversal without recursion and without stack!
Structure Member Alignment, Padding and Data Packing
Intersection point of two Linked Lists
Lowest Common Ancestor in a BST.
Check if a binary tree is BST or not
Sorted Linked List to Balanced BST

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    // If a part of this segment overlaps with the given range
    int mid = getMid(ss, se);
    return getSumUtil(st, ss, mid, qs, qe, 2*index+1) +
           getSumUtil(st, mid+1, se, qs, qe, 2*index+2);
}

/* A recursive function to update the nodes which have the given index
   their range. The following are parameters
    st, index, ss and se are same as getSumUtil()
    i    --> index of the element to be updated. This index is in inpu
   diff --> Value to be added to all nodes which have i in range */
void updateValueUtil(int *st, int ss, int se, int i, int diff, int ind
{
    // Base Case: If the input index lies outside the range of this se
    if (i < ss || i > se)
        return;

    // If the input index is in range of this node, then update the va
    // of the node and its children
    st[index] = st[index] + diff;
    if (se != ss)
    {
        int mid = getMid(ss, se);
        updateValueUtil(st, ss, mid, i, diff, 2*index + 1);
        updateValueUtil(st, mid+1, se, i, diff, 2*index + 2);
    }
}

// The function to update a value in input array and segment tree.
// It uses updateValueUtil() to update the value in segment tree
void updateValue(int arr[], int *st, int n, int i, int new_val)
{
    // Check for erroneous input index
    if (i < 0 || i > n-1)
    {
        printf("Invalid Input");
        return;
    }

    // Get the difference between new value and old value
    int diff = new_val - arr[i];

    // Update the value in array
    arr[i] = new_val;

    // Update the values of nodes in segment tree
    updateValueUtil(st, 0, n-1, i, diff, 0);
}
```

Follow @GeeksforGeeks Subscribe

Recent Comments

akshay what happens to our code when we put it here??

Find if a given string can be represented from a substring by iterating the substring “n” times · 39 minutes ago

Akshay kabra I have done this in O(n) with no extra...

Find if a given string can be represented from a substring by iterating the substring “n” times · 42 minutes ago

Anurag Singh From node u, we visit it's adjacent UNVISITED...

Articulation Points (or Cut Vertices) in a Graph · 1 hour ago

po http://ideone.com/SVP9Iw iterative version you...

Write a C program to print all permutations of a given string · 1 hour ago

kanu i think recursion we can use for that.....

Oracle Interview | Set 11 (For Server Technology) · 1 hour ago

po here is the iterative version with time...

Write a C program to print all permutations of a given string · 1 hour ago

AdChoices

► Binary Tree

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
// Return sum of elements in range from index qs (quey start) to
// qe (query end).  It mainly uses getSumUtil()
int getSum(int *st, int n, int qs, int qe)
{
    // Check for erroneous input values
    if (qs < 0 || qe > n-1 || qs > qe)
    {
        printf("Invalid Input");
        return -1;
    }

    return getSumUtil(st, 0, n-1, qs, qe, 0);
}

// A recursive function that constructs Segment Tree for array[ss..se]
// si is index of current node in segment tree st
int constructSTUtil(int arr[], int ss, int se, int *st, int si)
{
    // If there is one element in array, store it in current node of
    // segment tree and return
    if (ss == se)
    {
        st[si] = arr[ss];
        return arr[ss];
    }

    // If there are more than one elements, then recur for left and
    // right subtrees and store the sum of values in this node
    int mid = getMid(ss, se);
    st[si] =  constructSTUtil(arr, ss, mid, st, si*2+1) +
              constructSTUtil(arr, mid+1, se, st, si*2+2);
    return st[si];
}

/* Function to construct segment tree from given array. This function
   allocates memory for segment tree and calls constructSTUtil() to
   fill the allocated memory */
int *constructST(int arr[], int n)
{
    // Allocate memory for segment tree
    int x = (int)(ceil(log2(n))); //Height of segment tree
    int max_size = 2*(int)pow(2, x) - 1; //Maximum size of segment tre
    int *st = new int[max_size];

    // Fill the allocated memory st
    constructSTUtil(arr, 0, n-1, st, 0);

    // Return the constructed segment tree
```

▶ Java Programming
▶ Java Code
AdChoices
▶ Keys Code
▶ PHP Example
▶ C++ Code
AdChoices
▶ A Tree Class
▶ C++ Example
▶ Segmentation

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
    return st;
}

// Driver program to test above functions
int main()
{
    int arr[] = {1, 3, 5, 7, 9, 11};
    int n = sizeof(arr)/sizeof(arr[0]);

    // Build segment tree from given array
    int *st = constructST(arr, n);

    // Print sum of values in array from index 1 to 3
    printf("Sum of values in given range = %d\n", getSum(st, n, 1, 3))

    // Update: set arr[1] = 10 and update corresponding segment
    // tree nodes
    updateValue(arr, st, n, 1, 10);

    // Find sum after the value is updated
    printf("Updated sum of values in given range = %d\n",
                                              getSum(st, n, 1, 3))

    return 0;
}
```

Output:

```
Sum of values in given range = 15
Updated sum of values in given range = 22
```

Time Complexity:
Time Complexity for tree construction is O(n). There are total 2n-1 nodes, and value of every node is calculated only once in tree construction.

Time complexity to query is O(Logn). To query a sum, we process at most four nodes at every level and number of levels is O(Logn).

The time complexity of update is also O(Logn). To update a leaf value, we process one node at every level and number of levels is O(Logn).

Segment Tree | Set 2 (Range Minimum Query)

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

References:
http://www.cse.iitk.ac.in/users/aca/lop12/slides/06.pdf

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

Bridge Java & .NET

Connect anything Java to .NET,
Connect anything .NET to Java

Related Topics:

- Binary Indexed Tree or Fenwick tree
- How to Implement Reverse DNS Look Up Cache?
- Given n appointments, find all conflicting appointments
- Perfect Binary Tree Specific Level Order Traversal
- Print Nodes in Top View of Binary Tree
- K Dimensional Tree
- Convert a Binary Tree to Threaded binary tree
- Serialize and Deserialize an N-ary Tree

Tags: Advance Data Structures, Advanced Data Structures, SegmentTree

51 Tweet 3

Writing code in comment? Please use **ideone.com** and share the link here.

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

64 Comments GeeksforGeeks

Sort by Newest

Join the discussion…

Guest • 2 months ago

sdf

• Reply • Share ›

srihari • 2 months ago

In getSumUtil function,
the first condition
why qs<=ss is taken instead of qs==ss

• Reply • Share ›

anubis • 2 months ago

to construct size of segment tree, 2*2^Ceiling(log2(n))-1: this will result a size
n=6
log2(n) = 2.58, ceiling of (2.58) = 3, i.e. 2*2^3-1 = 15. How come it gives 1?

• Reply • Share ›

helper • 3 months ago

this is quite interesting. we were taught this in class but i never knew impleme
implementation. thanks for sharing.

• Reply • Share ›

sree_ec • 3 months ago

while constructing the segement tree, size of segment tree can be calculated

In this example, it is complicated by using 2*(2^log2(n)) -1

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

where 2^log2(n) = n

3 ^ | v • Reply • Share ›

Ratan Roy ↪ sree_ec • 3 months ago

The formula is not 2*2^log2(n) -1. It is 2*2^Ceiling(log2(n)) -1. The reas
the same level. if they were, then the formula 2*2^log2(n) -1 will give th
representation.

1 ^ | v • Reply • Share ›

DS+Algo=Placement • 6 months ago

How is the complexity become O(n) for updation if we use array storing the su
'Another Solution'?

^ | v • Reply • Share ›

Boris Marchenko ↪ DS+Algo=Placement • 6 months ago

If you have array of 10 elements, you should store sum array of 10 ele
5, in the sum array you should update 5 elements. If we will update ele
number of elements to update in the sum array is N/2 (10 for element #
complexity is linear - O(N/2) = O(N).

^ | v • Reply • Share ›

DS+Algo=Placement ↪ Boris Marchenko • 6 months ago

Thanx

^ | v • Reply • Share ›

vipinkaushal • 6 months ago

i think the size allocated should be pow(2,h)-1 where h=ceil(log2(n))
because for last we have not to mark null child
please clear my doubt
thanks

^ | v • Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Ratan Roy ↪ vipinkaushal • 3 months ago

it is pow(2,h)-1 if root is considered to have height of 1.

it is 2*pow(2,h)-1 if root is considered to have height of 0.

Note: ceil(log2(n)), where n is number of leaves, gives a height that co

Reply • Share ›

ryan ↪ vipinkaushal • 5 months ago

no it should be 2*pow(2,h)-1, because (n-1) space is required to store
oru can say n-1 will be non leaf node and n will be leaf node

Reply • Share ›

matheus • 7 months ago

I think that the function constructSTUtil has a little problem, when we get on th
this operation: i*2+2 we get outside of the array on the next function call.

Reply • Share ›

ryan ↪ matheus • 5 months ago

no "st" array is sufficient enough

Reply • Share ›

anonymous • 7 months ago

// Update the values of nodes in segment tree
updateValueUtil(st, 0, n-1, i, diff, 0); i think is wrong
it should be updateValueUtil(st, 0, n-1, i, diff, i); since we update child nodes of

Reply • Share ›

gaurav • 8 months ago

Shouldn't 'if (qs <= ss && qe >= se)' in getSumUtil() function be 'if (qs >= ss &

Reply • Share ›

prashant • 9 months ago

void update(tnode* root int arr[] int low int high int ind int val)

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

void update(tnode *root,int arr[],int low,int high,int ind,int val)

{

if(low==high)

{

root->data=val;

return;

}

int mid=(low+high)/2;

if(ind>mid)

update(root->rchild,arr,mid+1,high,ind,val);

if(ind<=mid)

see more

Reply • Share ›

prashant • 9 months ago

note the crucial points
1-segment tree is not complete binary tree
2-the left and right subtree are divided based on middle values
3-the sum and update code are similar to bianry search code so its implemen
for update process
construction is simple ...just recursively go down dividing the index into 2 halve
node data and during unwinding phase adjust root->data
for range sum
if ind1>mid then just move to right subtree
if ind2<=mid move to left
else split the range index into 2 halves

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
int sum(tnode* root,int low,int high,int l,int h)

{

if((low==l)&&(high==h))

return root->data;
```

see more

Reply • Share ›

dmr • 9 months ago

We can use segment trees for finding sum in a given range. But how can this
in a given range in O(logn) time? This is a question in References link (
http://www.cse.iitk.ac.in/user... given above.

Reply • Share ›

prshant jha • 10 months ago

here is my update version in 0(logn) complexity with much simpler implement
http://ideone.com/SppdWT

3 • Reply • Share ›

The_Geek prshant jha • 5 days ago

IMO, It is not a proper implementation also. Because while creating tre
of elements. But while fetching sum you are again computing sum.The
So the implementation does not seem correct to me.
If anybdy can clarify here?

Reply • Share ›

The_Geek prshant jha • 5 days ago

How (arr+i) can work, while computing range query, because memory
So it is not guaranteed that (arr+i) will move root pointer to i-th node frc

Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

GOPI GOPINATH ➦ prshant jha • 9 months ago

will the complexity in your implementation be O(logn) ???? but to find tl
runtime ryt ????

^ | ˅ • Reply • Share ›

prashant jha • 10 months ago

```
#include<iostream>

using namespace std;

struct node

{

node* lchild;

int data;

node* rchild;

node()

{

lchild=NULL;

rchild=NULL;
```

see more

^ | ˅ • Reply • Share ›

prashant jha • 10 months ago

here is the implementation of the segment tree
http://ideone.com/JPDzqz

^ | ˅ • Reply • Share ›

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

nwoebcke • 10 months ago

Although this puts an additional O(log(n)) of memory on the heap, I think it mal
recursion a little easier to follow. Also it is C++. To preserve the C language, y
classes and rename the methods to functions with a struct pointer parameter

```
class Range {
  public:
   int start;
   int end;
   Range(s, e) : start(s), end(e) {}
   void init(s, e) {
      start = s;
      end = e;
   }
   inline bool isInside(Range *other) {
      return start >= other->start && end <= other->end;
   }
   inline bool isOutside(Range *other) {
      return start > other->end || end < other->start;
```

see more

Reply • Share ›

Puneet Jaiswal • a year ago

Would this work as tree implementation

```
public class SegmentTree {
public static class STNode {
int leftIndex;
int rightIndex;
int sum;
STNode leftNode;
```

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

```
STNode rightNode;
}

static STNode constructSegmentTree(int[] A, int l, int r) {
if (l == r) {
STNode node = new STNode();
node.leftIndex = l;
node.rightIndex = r;
node.sum = A[l];
```

see more

• Reply • Share ›

mallard • a year ago

sorry but i think the language of implementation of above code is C and i think getting error when i am trying to run this code.I am using codeblock.

• Reply • Share ›

Pranjal Ranjan mallard • 3 months ago

You can use malloc instead of new

• Reply • Share ›

Newbie90 • a year ago

What is the difference between the constructST and constructSTUtil functions constructSTUTil fucntion.

• Reply • Share ›

Ankur Sao Newbie90 • 9 months ago

ConstructST only allocates memory for the segment tree array and ca actually fills up the segment tree array. ConstructST than returns this a

• Reply • Share ›

Vu Duc Minh • a year ago

I do not think the procedure "updateValueUtil" is a good one (even it is correct)
should update from a leaf to the root; like the "constructSTUtil" procedure. In fa
"constructSTUtil". We only need one procedure for all two tasks.

Reply • Share ›

Adrian Carballo • a year ago

Hey, great tutorial, I wrote a python implementation here https://github.com/adı

Reply • Share ›

Avinash Ks • a year ago

Just one doubt, in SumUtil, isn't qs supposed to be greater than ss and qe les
of ss - se

Reply • Share ›

Denis • a year ago

Hi, you wrote : "size of memory allocated for segment tree will be 2*2^|log2n|-1
where I assume n is a number of leafs in the tree. This is seems not to
be true using your example of a segment tree : "{1,3,5,7,9,11}",
where n=6. Thus using 2*2^|log2n|-1 size of memory allocated is 7, which is n
are 11 nodes. Suppose n should be replaced
on the (2*n-1) in your expression.

Reply • Share ›

Denis ➦ Denis • a year ago

Sorry, I've got it. You are using ceil() function in this case. So 2*2^ceil(l
tree size. I was thinking about floor() instead of ceil().

Reply • Share ›

Vishnu Kamavaram ➦ Denis • 9 days ago

if for suppose n=30
ceil(log(30))=2.0
size= 2*2^2-1 =7(i think its wrong)

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

ıт ı wrong do correct me !!!

^ | ⌄ • Reply • Share ›

Denis • a year ago

Hi, you wrote : "size of memory allocated for segment tree will be ", where I as
tree. This is seems not to be true using your example of a segment tree : "{1,3
where n=6. Thus using size of memory allocated is 7, which is not true becau
Suppose n should be replaced on the (2*n-1) in your expression.

^ | ⌄ • Reply • Share ›

denial • a year ago

@geeksforgeeks:
Change suggestion in the paragraph "Query for Sum of given range". It should

```
int getSum(node, l, r)
{
   if range of node is within l and r
        return value in node

   if range of node is completely outside l and r
        return 0

    return getSum(node's left child, l, r) +
           getSum(node's right child, l, r)
}
```

let me know if I'm wrong. :)

^ | ⌄ • Reply • Share ›

denial • a year ago

@geeksforgeeks
Change suggestion in the paragraph "Query for Sum of given range" above :

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

You written it as :

```
int getSum(node, l, r)
{
   if range of node is within l and r
        return value in node
   else if range of node is completely outside l and r
        return 0
   else
    return getSum(node's left child, l, r) +
           getSum(node's right child, l, r)
}
```

should be changed to this:

see more

^ | ⌄ • Reply • Share ›

Prakhar Jain • 2 years ago

Time Complexity of query is O(log n) because we process at most "4 nodes"
nodes" which is wrong. For example take range [1-3] in your example and mal
function, you will see there are at most 4 nodes at each level.
Even it is also given in the iitk link you have given at the end.

```
/* Paste your code here (You may delete these lines if not writing co
```

1 ^ | ⌄ • Reply • Share ›

GeeksforGeeks ➦ Prakhar Jain • 2 years ago

@Prakhar Jain: Thanks for pointing this out. we have updated the post

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Reply • Share ›

Prakhar Jain GeeksforGeeks • 2 years ago

Also, to update a leaf we process "two nodes" at each level, no

```
/* Paste your code here (You may delete these lines if r
```

Reply • Share ›

kartik • 2 years ago

How do we do the updation if we have to update more than 2 values

like we have to increase all number in range a to b by 2

how our update function do this in O(log(n))

can any body plz help

```
/* Paste your code here (You may delete these lines if not writing coo
```

Reply • Share ›

Sumanth Bandi • 2 years ago

&#039&#039Since the tree is represented using array and relation between pa
maintained, size of memory allocated for segment tree will be 2*(2^ceil(log2n)
line.

Why not just 2*n - 1? What are the bad sequences of just allotting 2*n - 1 node
Anybody pls help..

Reply • Share ›

sumanth232 • 2 years ago

Since the tree is represented using array and relation between parent and chil

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

memory allocated for segment tree will be 2*(2^ceil(log2n)) - 1.

Why not just 2*n - 1 ? What are the bad sequences of just allotting 2*n - 1 nod
Anybody pls help..

Reply • Share ›

alveko sumanth232 • 2 years ago

The array must have enough elements to include a possible right-mos
most leaf increases with a step of power of 2. The size of (2*n-1) migh

```
// segment tree size (n is the number of elements in the input
//     (log2ceil(n))   is the level that can hold all distinct
//   2^(log2ceil(n))   is the number of elements at that level
// 2*2^(log2ceil(n))-1 is the total number of elements in the t
```

/ Alexander K.

Reply • Share ›

sumanth232 alveko • 2 years ago

Thanks.. that made it clear..

```
/* Paste your code here (You may delete these lines if r
```

Reply • Share ›

abhishek08aug • 2 years ago

Intelligent :D

Reply • Share ›

Ouditchya Sinha • 2 years ago

Very nicely explained! Thank You GeeksforGeeks... :) Just one question, wher

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

Very nicely explained! Thank You GeeksforGeeks... :) Just one question, whe
the value in array & construct another segment tree? It seems to be working h
won&#039t need update functions, would it be computationally costly? Will thi

^ | ⌄ • Reply • Share ›

Load more comments

Subscribe Add Disqus to your site Privacy

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com

@geeksforgeeks, **Some rights reserved** **Contact Us!** Powered by **WordPress** & **MooTools**, customized by geeksforgeeks team

open in browser PRO version Are you a developer? Try out the HTML to PDF API pdfcrowd.com