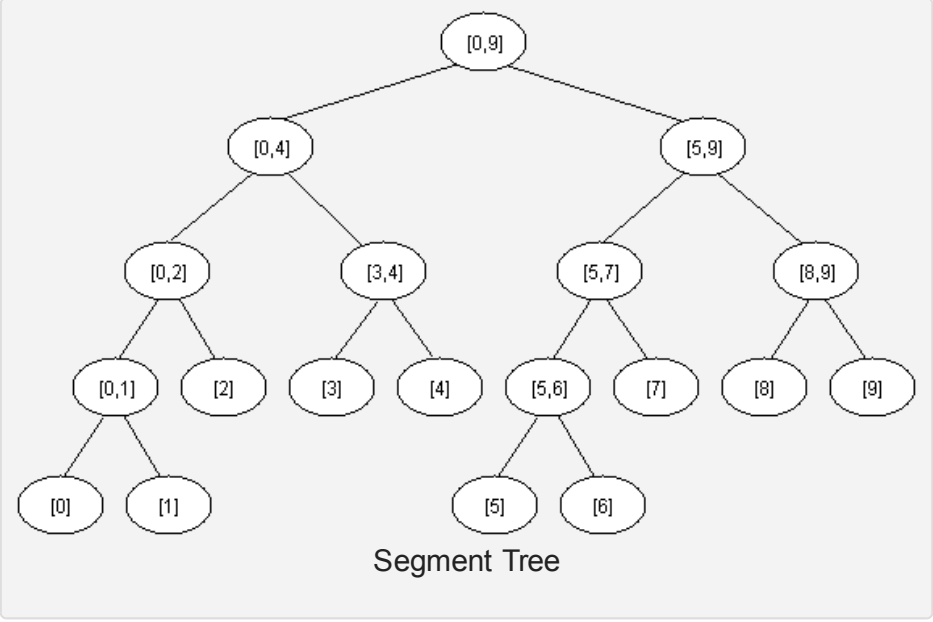
Segment Trees

with 18 comments

A segment tree is a heap-like data structure that can be used for making update/query operations upon array intervals in logarithmical time. We define the segment tree for the interval [i, j] in the following recursive manner:

- the first node will hold the information for the interval [i, j]
- if i<j the left and right son will hold the information for the intervals **[i, (i+j)/2]** and **[(i+j)/2+1, j]**

See the picture below to understand more :



Segment Tree

We can use segment trees to solve Range Minimum/Maximum Query Problems (RMQ). The time complexity is **T(N, log N)** where O(N) is the time required to build the tree and each query takes O(log N) time. Here's a C++ template implementation :

I am

A graduate from Institute of Technology,BHU. Coding, maths, number theory, oeis, music, reality shows, cs1.6 and sleeping, that pretty much sums up my life :)

Project Euler

Search

Email Subscription

Enter your email address to subscribe to this blog and receive notifications of new posts by email.
Join 264 other followers

Enter your email address

Sign me up!

```
#include<iostream>
using namespace std;
#include<math.h>

template<class T>
class SegmentTree
{
      int *A,size;
      public:
      SegmentTree(int N)
      {
           int x = (int)(ceil(log2(N)))+1;
           size = 2*(int)pow(2,x);
           A = new int[size];
           memset(A,-1,sizeof(A));
      }
      void initialize(int node, int start,
                      int end, T *array)
      {

           if (start==end)
              A[node] = start;
           else
           {
               int mid = (start+end)/2;
               initialize(2*node,start,mid,array);
               initialize(2*node+1,mid+1,end,array);
               if (array[A[2*node]]<=
                      array[A[2*node+1]])
                  A[node] = A[2 * node];
               else
                   A[node] = A[2 * node + 1];
           }
      }
      int query(int node, int start,
                int end, int i, int j, T *array)
      {
          int id1,id2;
          if (i>end || j<start)
             return -1;

          if (start>=i && end<=j)
             return A[node];

          int mid = (start+end)/2;
          id1 = query(2*node,start,mid,i,j,array);
```

```
            id2 = query(2*node+1,mid+1,end,i,j,array);

            if (id1==-1)
               return id2;
            if (id2==-1)
               return id1;

            if (array[id1]<=array[id2])
               return id1;
            else
                return id2;
        }
};

int main()
{
     int i,j,N;
     int A[1000];
     scanf("%d",&N);
     for (i=0;i<N;i++)
         scanf("%d",&A[i]);

     SegmentTree<int> s(N);
     s.initialize(1,0,N-1,A);
     while (scanf("%d%d",&i,&j)!=EOF)
       printf("%d\n",A[s.query(1,0,N-1,i-1,j-1,A)]);
}
```

Resources:

1. Topcoder Tutorial

NJOY!
-fR0D

Recent Comments

My Tweets

Error: Twitter did not respond. Please wait a few minutes and refresh this page.

Visitors Location

Rate this:

26 3 Rate This

Share this:

Related

Binary Indexed Tree (BIT)
In "Programming"

Range Minimum Query
In "Algorithm"

Recurrence Relation and Matrix Exponentiation
In "Programming"

Written by fR0DDY
September 15, 2009 at 8:21 PM

Posted in Programming
Tagged with C, code, complexity, data, RMQ, Segment, structure, time, Tree

« Subsequence Frequency

Binary Indexed Tree (BIT) »

18 Responses

Subscribe to comments with RSS.

Hey, nice tut. If u add a tutorial for updating the values in segment tree, it will be great!

Boris
September 18, 2010 at 12:34 PM

Reply

---

when i five n= 800 in input then this code crashes.
please help

Anuj
October 13, 2010 at 8:53 AM

Reply

---

please add the nessary discription about n,A[i]. and also add the necassary printf statements . please reply immediately

berin
November 12, 2010 at 7:22 PM

Reply

Do you think this guy's your slave or what?

Andrés Mejía
November 14, 2010 at 8:55 PM

Reply

---

could u plzzz add a good tutorial on binary indexed tree as they r easy to code????

aayush kumar
June 9, 2011 at 4:49 PM

Reply

There already is a post on that
https://comeoncodeon.wordpress.com/2009/09/17/binary-indexed-tree-bit/

fR0DDY
June 9, 2011 at 8:30 PM

Reply

Hi I Don't Understand Whats the application of segment tree ? i mean if we can search the element in given range of sorted array in O(logn) then why we need such complex DS or m i missing sum-thing so do u mean we can find the elements in unsorted array in O(logn) is it so .?? as Heap can unsorted array as 5 4 3 1 2 isn't it .?? also please explain the in detail the initialize & query part & also write update part as you have mentioned..i am really interest in algorithms & so i wants to know what we can do with segment tree once you will reply my question i will really look & analyze it…i mean really really interested & appreciate ur attempt.

i mean when i m giving input for i & j 0 ,5 or i=0 & j=1 to 9 for N=10 array then i am getting output of query is 0 m not getting what exactly query function is doing ?? whats the purpose of it does it s giving element in range or its searching particular element & returning that element.

Reply ASAP.

Algoseekar
June 12, 2011 at 10:08 PM

It can be used to find maximum/minimum element in a range of an unsorted array.
Also between the queries also, you can update any element.

fR0DDY
June 15, 2011 at 10:27 PM

Bugs in the code above.
-> log is logarithm to the base 10, whereas log to the base 2 should be used.
-> only the position should be returned in line numbers 50,52,55,57.

To verify the bug run the code with the following input
N = 8
array = { 1, 2, 0, -1, 5, 5, 5, 5 }
first i,j -> 1 2 (gives correct output of 0)

mozzak
October 1, 2011 at 12:43 PM

second i,j -> 3 4 (gives incorrect output of 0)

Reply

This is the output for this case- remember output is minimum element 🙂

8
1 2 0 -1 5 5 5 5
1 2
1
3 4
-1

Reply

shashank jain
August 2, 2012 at 7:25 PM

It would be better if in the tree initialization N = 2^X, you would get faster solution. Your current solution would get TL on some test cases… I don't remeber testcases, but I promise you that N should be equal 2^x 🙂

Reply

vilvler
February 27, 2012 at 3:02 PM

Why would it be faster if you are adding more nodes to the tree?

Reply

Andrés Mejía
February 28, 2012 at 5:38 AM

Nice tutorial, I also wanted to know about updation of segment trees. How is it achieved? Can you explain a little more?

Reply

aman
June 18, 2012 at 8:54 AM

hey how to implement if i need all intervals with 0<=i<j<n
like for eg in the above tree i need to fint the max from the interval [1,8]

Reply

aichemzee
December 9, 2012 at 2:55 AM

I beleive the time complexity of construction the segment tree is $O(n\log n)$.

Macropodus
February 23, 2013 at 9:51 PM

Reply

There's a small bug, not so important yet proves pain in neck if tested under certain input ranges. For very large range say 1 to 100000. The Query function goes so deep in recursion that it exceeds the recursion depth & hence will result as "Segmentation Fault". I've tried it locally on my machine and on online competition too to verify.

Ravi Ojha
May 12, 2013 at 1:59 PM

Reply

[…] wcipeg.com/wiki/Segment_tree comeoncodeon.wordpress.com/2009/09/15/segment-trees/ letuskode.blogspot.com/2013/01/segtrees.html […]

Segment Trees | Sport CoderSport Coder
December 5, 2013 at 1:25 AM

Reply

[…] en.wikipedia.org/wiki/Segment_tree wcipeg.com/wiki/Segment_tree comeoncodeon.wordpress.com/2009/09/15/segment-trees/ letuskode.blogspot.com/2013/01/segtrees.html p–np.blogspot.com/2011/07/segment-tree.html […]

Segment Trees - Lazy Updates
December 10, 2013 at 10:03 PM

Reply

Leave a Reply

Enter your comment here...