

Segment Trees

Longest Non Decreasing Subsequence (Revisited)

1	2	5	2	8	6	3	6	9	7
---	---	---	---	---	---	---	---	---	---

Length of LNDS ending at i^{th} Loc

1	2	3	3	4	4	4	5	6	6
---	---	---	---	---	---	---	---	---	---

This runs in $O(n^2)$. Can we do better?

Challenge: Given a value of $a[i]$ we need to find the maximum length possible among all subsequences ending at any value less than equal to $a[i]$ quickly.

Possible with small change in algorithm and usage of STL, refer to the link
http://www.algorithmist.com/index.php/Longest_Increasing_Subsequence

We will look at another way of doing it in a restricted setting when all elements in the initial array are not really large say bounded by $1e6$.

Find the Max value among the first i elements of an array

3	4	1	7	13	5	21	6	23	16
---	---	---	---	----	---	----	---	----	----

Naïve Approach: Just scan till the ith index and output the maximum.

Cost of operations:

Finding maximum $O(1)$ Updating an element in initial array $O(n)$

Another Approach: Maintain another array,
each index storing the maximum value up to that index.

3	4	4	7	13	13	21	21	23	23
---	---	---	---	----	----	----	----	----	----

Cost of operations:

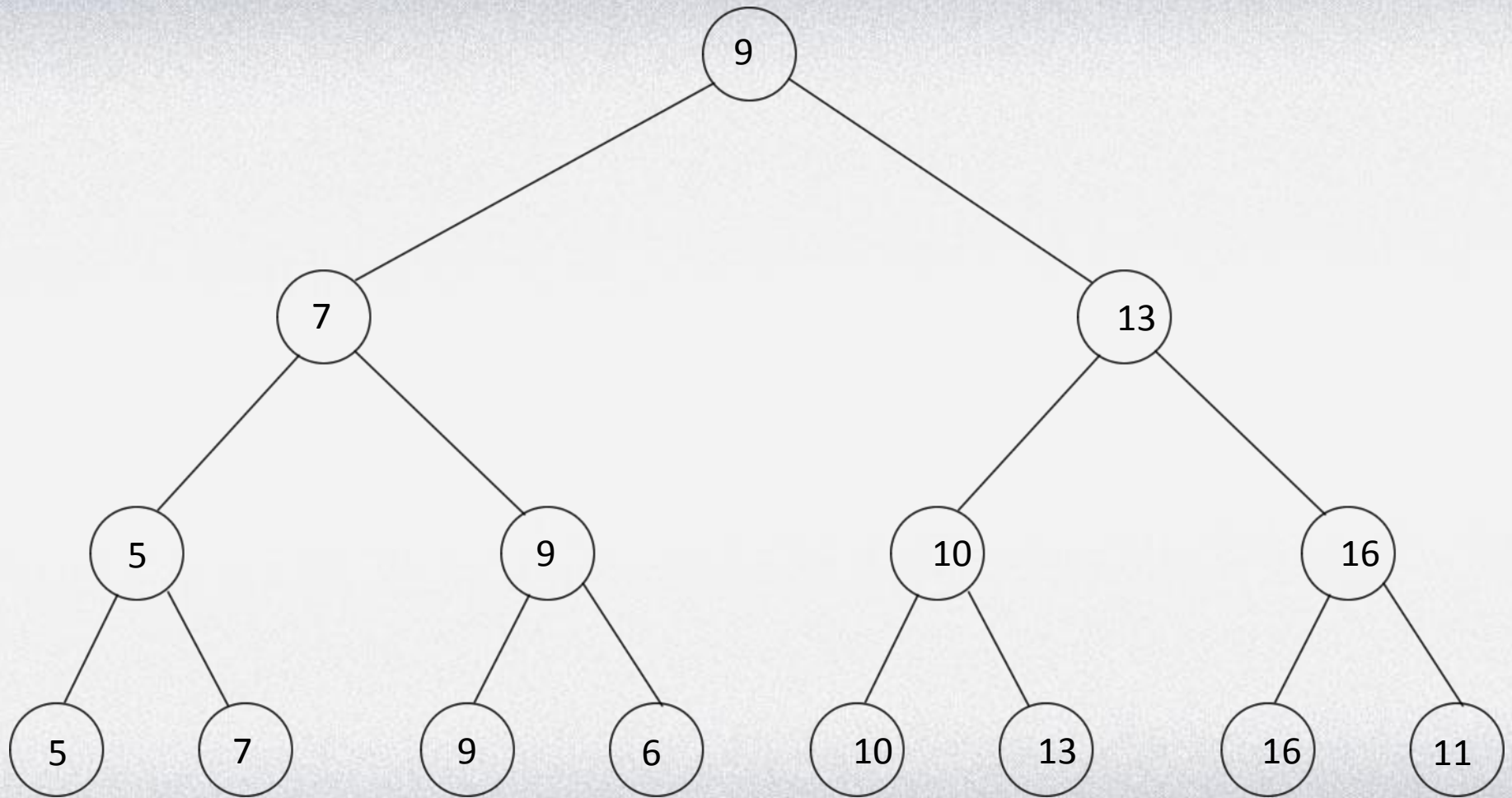
Finding maximum $O(1)$ Updating an element in initial array $O(n)$

This works well if the number of query operations are large and very few updates

If the number of query and updates are equal
then it is as good/bad as a naïve way of doing it.

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

A Basic Segment Tree

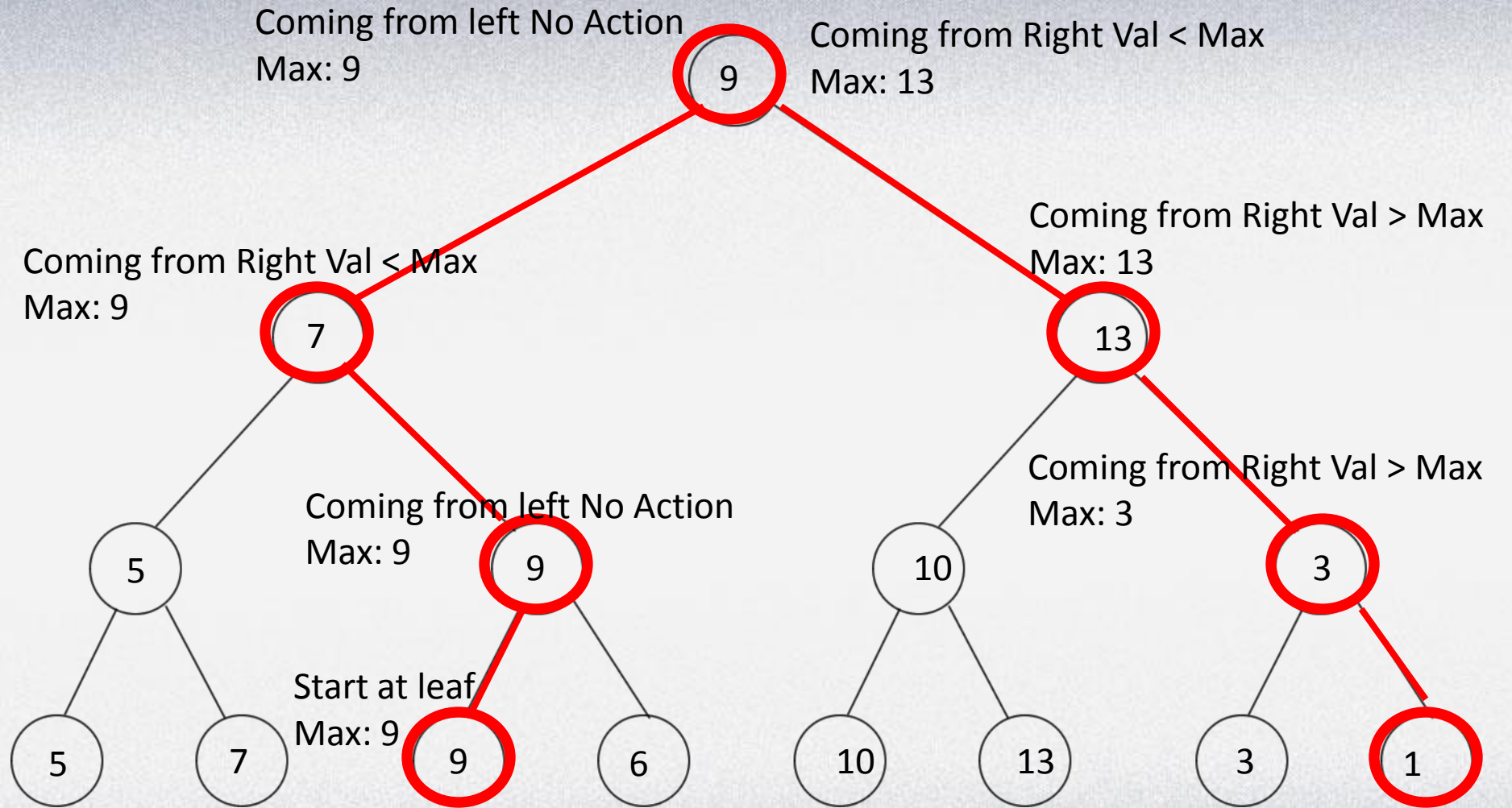


Leaf Nodes are the elements in the array.

Each internal node represents some merging of the leaf nodes

Fill each node with the maximum value in the left sub array.

Querying



To find the maximum value stored up to the i^{th} index do the following

Start at the i^{th} leaf with max set to the value in the leaf, keep traversing till the root.

If you had reached the parent from the left child don't take any action
otherwise if the max value in the left sub-tree is greater than max update max

- Updating an element in an array and precomputing – Just the reverse of querying.

Update the i^{th} leaf keep with value 'v' traversing till the root.

If you had reached the parent from the right child don't take any action otherwise if the value in the node is less than 'v' update it with 'v'

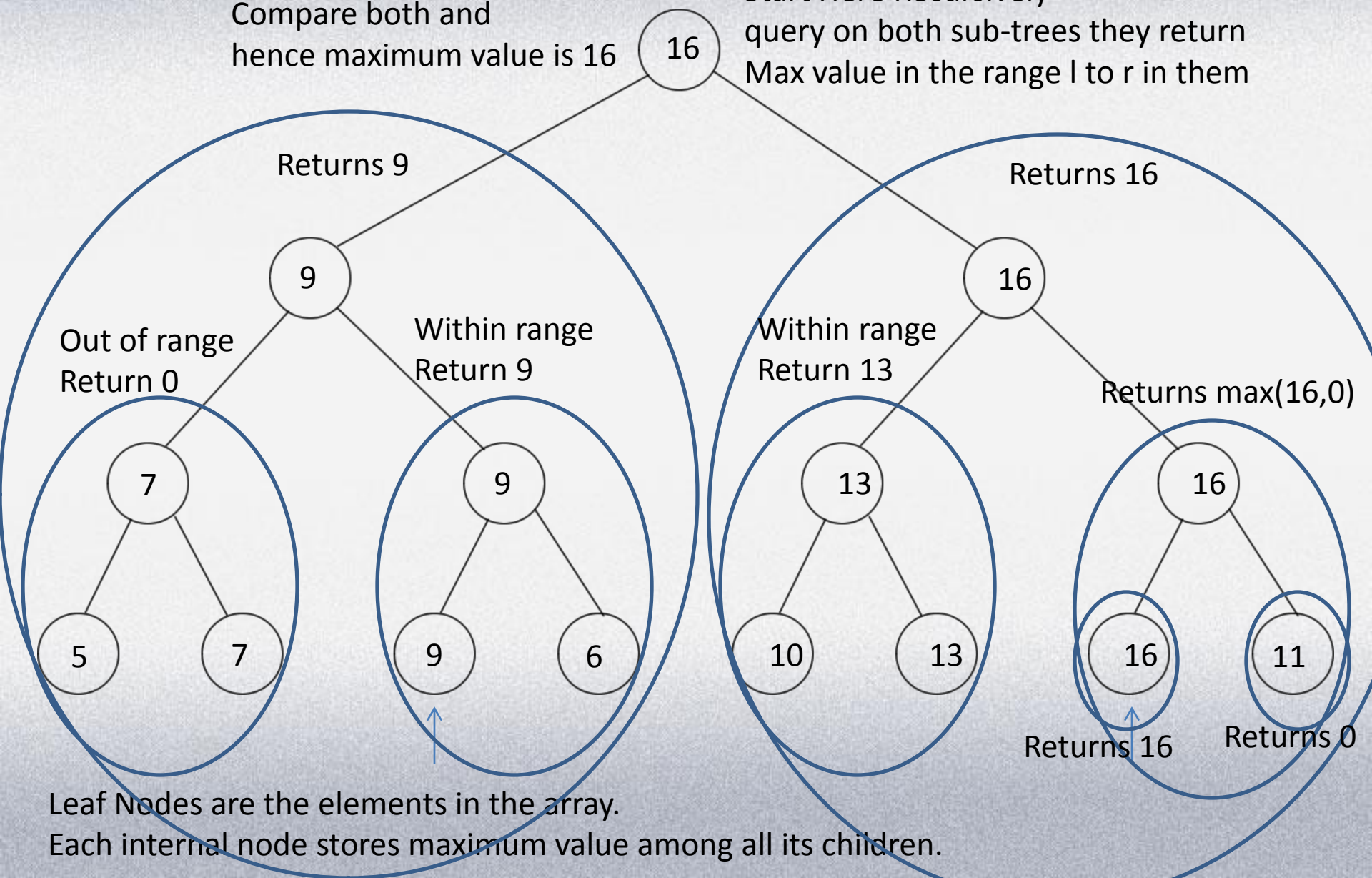
- For each operation we need to travel till the root of the tree.
- Height of tree is bounded by $O(\log n)$ hence each operation can be done in $O(\log n)$
- If the query range is from i^{th} index to the end of the array just do the reverse of what we had done
- What if the query range is any segment in between the array?

Segment Tree

Compare both and
hence maximum value is 16

Start Here Recursively

query on both sub-trees they return
Max value in the range l to r in them



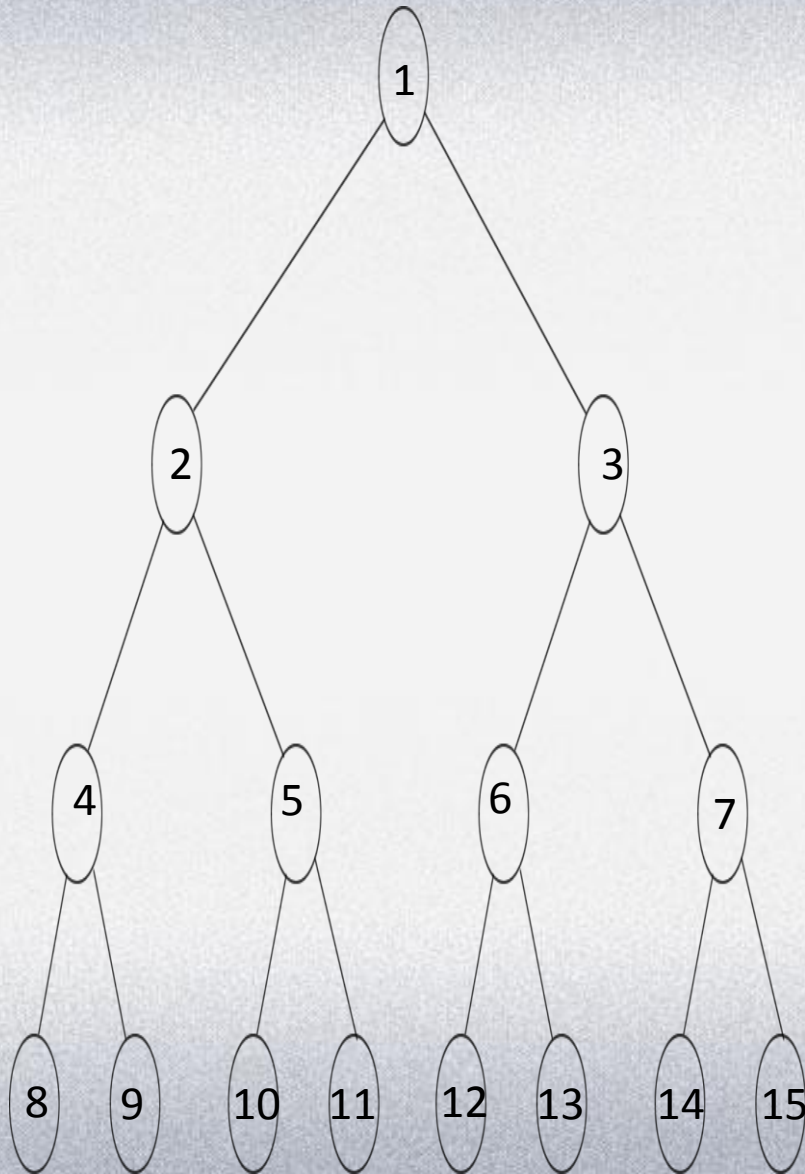
Querying

- Query(root,l,r)

```
query(node,l,r){  
    if range of node is within l and r  
        return value in node  
    else range of node is completely outside l and r  
        return 0  
    else  
        return max(query(left-child,l,r),query(right-child,l,r))  
}
```

- range of node is the [left most leaf,right most leaf] in the sub-tree rooted at node.
- How to represent the tree
- How to check if the range is between left and right
- ~~I think it will be pretty tough to code this~~
- It is very simple to code a segment tree
- Demo

Representation of the tree



- Number each node in the tree level by level
- Observe that for each node the left child is $2*i$ and right child is $2*i+1$
- For each node the parent is $i/2$
- Just use an array to represent the tree, operate on indices to access parents and children
- For ease of access ignore the 0-indexed element in the array

- Updates as well as queries on intervals (On Board)
- Read http://translate.googleusercontent.com/translate_c?act=url&hl=en&ie=UTF8&prev=t&rurl=translate.google.com&sl=auto&tl=en&twu=1&u=http://e-maxx.ru/algo/segment_tree&usg=ALkJrhgDDakTy9AeCVpP7dGwJoPxR--qSw before attempting any of the problems
- Read about Binary indexed trees <http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=binaryIndexedTrees>

Practice Problems

- <http://www.spoj.pl/problems/GSS1/>
- <http://www.spoj.pl/problems/GSS3/>
 - <http://pclub.in/index.php/wpc-archives/16-kodefest-solutions/89-problem-g>
- <http://www.spoj.pl/problems/HORRIBLE/>
- <http://www.spoj.pl/problems/TEMPLEQ/>
- <http://www.codechef.com/problems/FLIPCOIN/>
- <http://www.codechef.com/problems/MULTQ3>