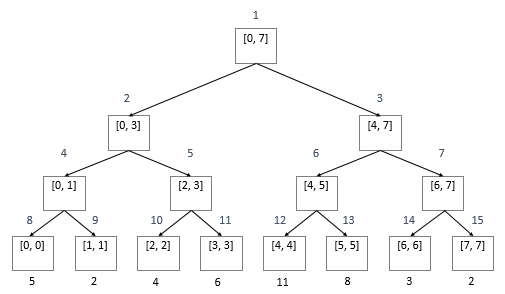
Segment Tree



Segment Tree Data Structure



Base on different problems, we can add extra instance variables in the Segment Tree Node class.

Start – start index (inclusive)

End – end index (inclusive)

Left – left sub-tree

Right – right sub-tree

Segment Tree Build

1. No extra instance variables (LintCode 201)



1. Given an array, implement a build method with a given array, so that we can create a corresponding segment tree with every node value represent the corresponding interval max value in the array, return the root of this segment tree. (LintCode 439)





Segment Tree Query

1. Design a query method with three parameters root, start and end, find the maximum number in the interval [start, end] by the given root of segment tree. (LintCode 202)



1. For an array, we can build a Segment Tree for it, each node stores an extra attribute **count** to denote the number of elements in the array which value is between interval start and end. (The array may not fully filled by elements)

Design a **query** method with three parameters **root**, **start** and **end**, find the number of elements in the in array's interval [start, end] by the given root of value Segment Tree. (LintCode 247)

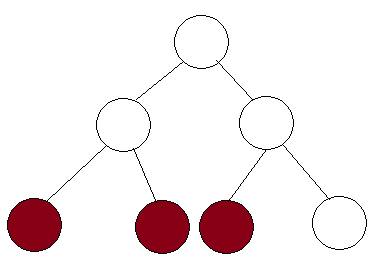


Query Time Complexity: O(logn)

The claim is that there are at most 2 nodes which are expanded at each level.

We will prove this by contradiction.

Consider the segment tree given below.



Let's say that there are 33 nodes that are expanded in this tree. This means that the range is from the left most colored node to the right most colored node. But notice that if the range extends to the right most node, then the full range of the middle node is covered. Thus, this node will immediately return the value and won't be expanded. Thus, we prove that at each level, we expand at most 2 nodes and since there are logn levels, the nodes that are expanded are 2⋅logn = Θ (logn).

(Reference: https://cs.stackexchange.com/questions/37669/time-complexity-proof-for-segment-tree-implementation-of-the-ranged-sum-problem/39594#39594?newreg=66ea671b3e0245179116ef52fbe753cf)

Segment Tree Modify

For a **Maximum Segment Tree**, which each node has an extra value **max** to store the maximum value in this node's interval.

Implement a **modify** function with three parameter **root**, **index** and **value** to change the node's value with ***[start, end] = [index, index]*** to the new given value. Make sure after this change, every node in segment tree still has the **max** attribute with the correct value. (LintCode 203)

