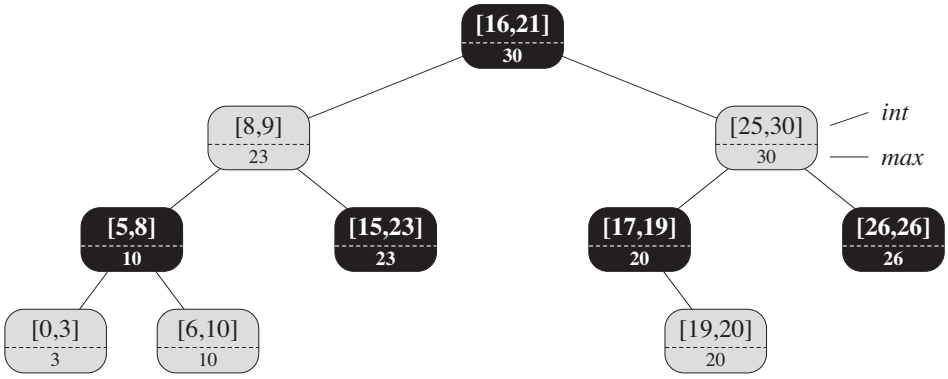
Problem 1-1. (Interval Tree)

(a) Show how INTERVAL-SEARCH$(T.root, [6,7])$ operates on the red-black tree T in the following figure.



(b) Modify pseudocode LEFT-ROTATE to operate on nodes in an interval tree and update the max attributes in $O(1)$ time.

(c) Describe an efficient algorithm that, given an interval i, returns an interval overlapping i that has the minimum low endpoint, or $T.nil$ if no such interval exists.

(d) Given an interval tree T and an interval i, describe how to list all intervals in T that overlap i in $O(\min(n, k \lg n))$ time, where k is the number of intervals in the output list.

Problem 1-2. (Amortized Analysis)

(a) Suppose we perform a sequence of n operations on a data structure in which the ith operation costs i if i is an exact power of 2, and 1 otherwise.

Use aggregate analysis to determine the amortized cost per operation.

(b) Redo Part (a) using an accounting method of analysis.

(c) Redo Part (a) using a potential method of analysis.