CIS 313

Intermediate Data Structures

Final • December 10, 2019 • 10:15–12:15

Name _____

Use the space provided for each answer. The points for each item are shown in brackets. You may use one page of handwritten notes (8.5 x 11, both sides). Good luck!

TIP 1: Some questions are trickier than others. Do the problems you know how to do first, then come back for the rest.

TIP 2: If a question is ambiguous or confusing, please ask about it.

1. [3] Using the *formal definition*, prove that $n^2 + n + 1$ is $O(n^2)$.
   NOTE: To receive full credit, you must show that the definition holds, not simply state values of c and n_0.

2. [3] Consider the following piece of code that takes as input a positive integer $n > 0$:

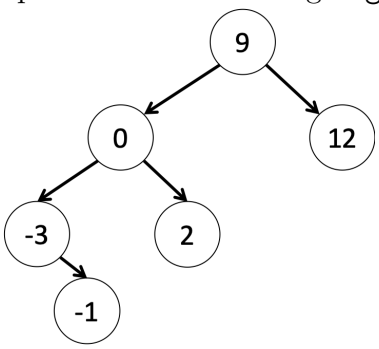
```
i = 0
j = 1
while i < n:
    k = 1
    while k < j:
        k = k * 2
    j = j * 4
    i = i + 1
```

Using big-O notation, specify the runtime of this code as a function of n. Provide the tightest bound you can. Briefly justify your answer.

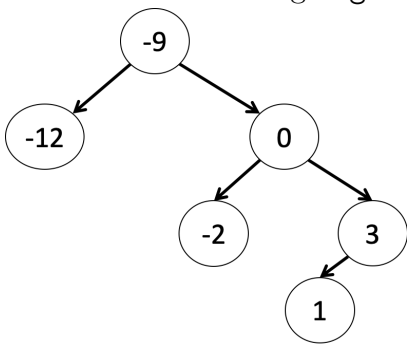
3. [4] Write pseudo-code for a `negate()` method, which negates the data in a binary search tree. Your final tree may have a different structure, but it must still preserve the binary search tree ordering (i.e., for each node, values in left subtree $<$ value at node $<$ values in right subtree). For full credit, your method should run in time $O(n)$ (but you don't need to prove this).

   The root of the tree `T` is `T.root`, and each node `n` has fields `n.left` and `n.right` to point to its left and right children respectively.

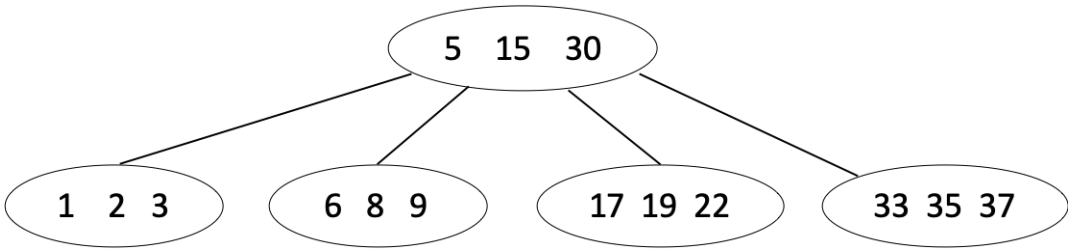
   Example BST before calling `negate`:



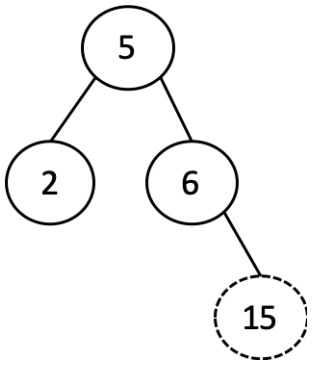
   Same BST after calling `negate`:



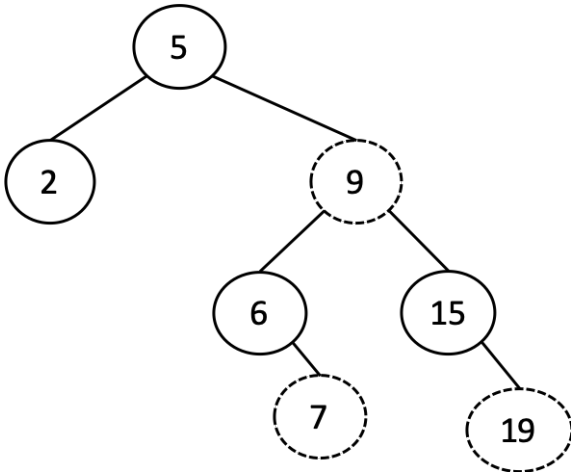
4. [3] Show the structure of the following b-tree with minimum degree 3 after inserting 11, 13, 20, 21, 18, 39, 41, 43, 4. To save time and space, you may omit intermediate steps – just draw the final tree structure. (You may find it helpful to use the back of the paper as scratch space.)



5. [3] Show the structure of the following red-black tree after inserting 24, 25, 26, and 27. Red nodes are indicated by dotted lines. You must clearly indicate which nodes are red in your final answer.



(a) [2] Show the structure of the following red-black tree after deleting 6. Red nodes are indicated by dotted lines. You must clearly indicate which nodes are red in your final answer.



(b) [2] Show the structure of the original red-black tree (before deletion of 6) after deleting 2. Red nodes are indicated by dotted lines. You must clearly indicate which nodes are red in your final answer.

6. [6] Suppose we have a school with n teachers and n^2 students. The students have student numbers $0, 1, 2, \ldots, (n^2 - 1)$. The teachers of the school vote for the "most outstanding student" award, and there are n votes.

   Describe how to determine which of the n^2 students received the largest number of the votes (if two or more students tie for the highest number of votes, you can just pick one winner arbitrarily). The votes are given to you in an array $V[0, 1, \ldots, (n-1)]$, and for each i, $0 \leq V[i] < n^2$.

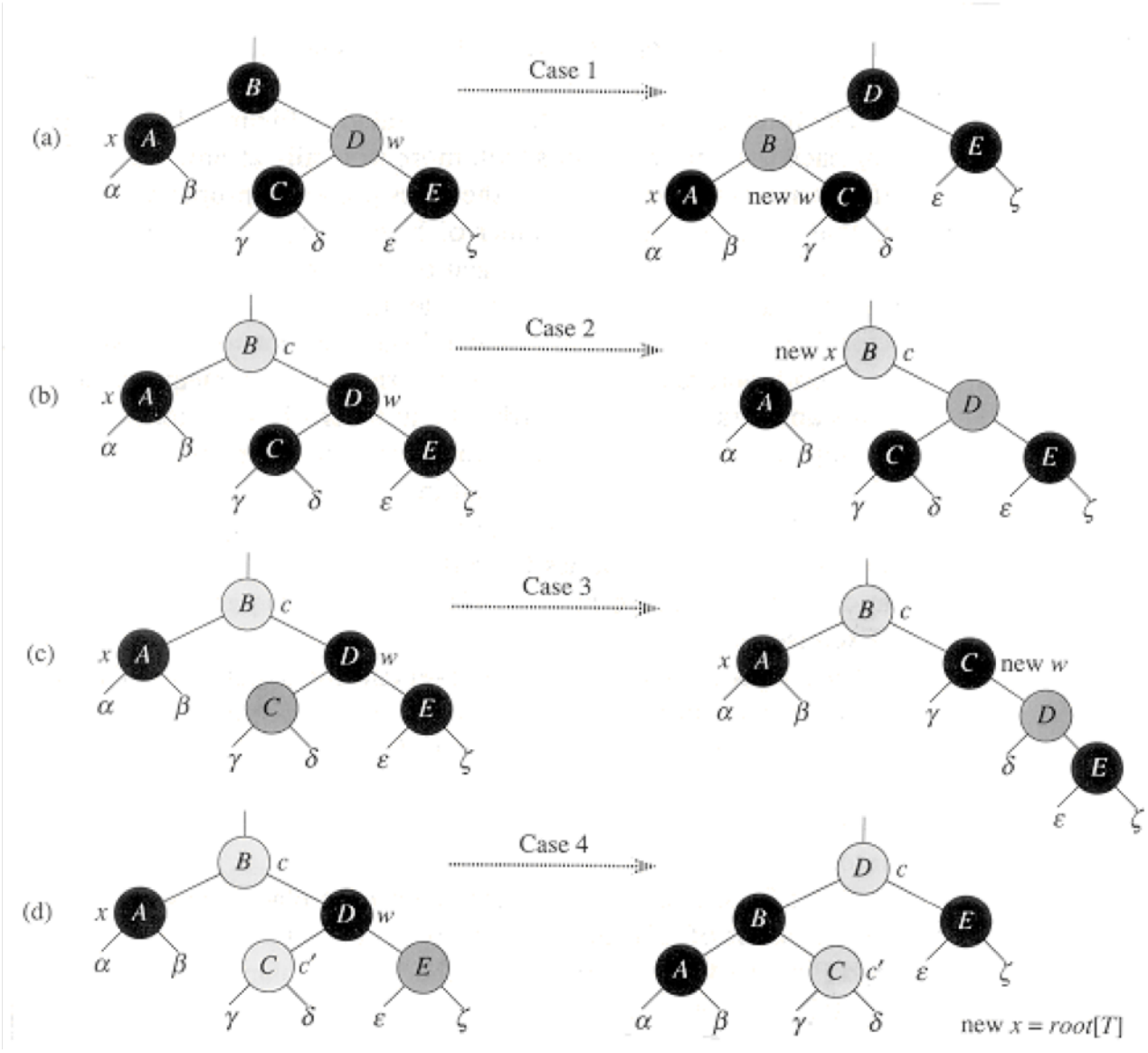
   For full credit, you must provide a method that runs in $O(n)$ time. Hint: This can be done using either radix sort or a hashtable.

7. Consider a hashtable with n buckets containing a total of n^2 items (in total, across all buckets in the entire data structure).

   Assume separate chaining is used to resolve collisions. What are the average case (assuming simple uniform hashing) and worst case running times for finding an item in this hash table? Briefly justify each answer.

   (a) [2] Average case:

   (b) [2] Worst case:


(a)
Case 1
B
x A
D w
α β
C E
γ δ ε ζ
D
B E
x A new w C ε ζ
α β γ δ
(b)
Case 2
B c
x A D w
α β C E
γ δ ε ζ
new x B c
A D
α β C E
γ δ ε ζ
(c)
Case 3
B c
x A D w
α β C E
γ δ ε ζ
B c
x A C new w
α β γ D
δ E
ε ζ
(d)
Case 4
B c
x A D w
α β C c′ E
γ δ ε ζ
D c
B E
A C c′ ε ζ
α β γ δ
new x = root[T]