ECE368 Fall2016 Homework #6

IMPORTANT: Write your user (login) ID at the TOP of EACH page. Also, be sure to ***read and sign*** the ***Academic Honesty Statement*** that follows:

> *"In signing this statement, I hereby certify that the work on this exercise is my own and that I have not copied the work of any other student while completing it. I understand that, if I fail to honor this agreement, I will receive a score of ZERO for this exercise and will be subject to possible disciplinary action."*
>
> `Printed Name:`
>
> `login:`
>
> `Signature:`

Please acknowledge those people who have helped you with this homework.

Question	Credits
1	
2	

1. Show that the tree height of a height-balanced binary search tree with `n` nodes is `O(log n)`. (Hint: Let `T(h)` denote the fewest number of nodes that a height-balanced binary search tree of height `h` can have. Express `T(h)` in terms of `T(h-1)` and `T(h-2)`. Then, find a lower bound of `T(h)` in terms of `T(h-2)`. Finally, express the lower bound of `T(h)` in terms of `h`.)

Solution:

$$\begin{aligned} T(h) &= T(h-1) + T(h-2) + 1 \\ &> 2T(h-2) \\ &> 2^2 T(h-4) \\ &> 2^{h/2} \end{aligned}$$

Let `n` be the number of nodes in a height-balanced tree of height `h`. Clearly, `n` $\geq$ `T(h) >` $2^{h/2}$. Therefore, `h < 2log n = O(log n)`.

2. The following diagram shows a height-balanced binary search tree. Perform the following operations: `insert 24`, `insert 19`, and `delete 50`. Use the procedure in Question 2 to perform node deletion. Perform rotation(s) to keep the binary search tree height-balanced after each operation.