

1. Recap: $\text{Delete}(e)$ algorithm (bottom-up way, not top-down)

- Step 1: Perform standard BST search for e .
- Step 2: Splay e to the root
- Step 3:
- Step 4:
- Step 5:

What is the time complexity for each step above?

2. **(True/False)** Another possible way to perform $\text{Delete}(e)$ would be to take the predecessor of e and place it directly at the root, without splaying it up through each level.

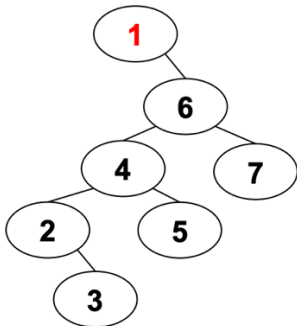
Why?

3. $\text{Find}(e)$ operation in a splay tree can be performed in two ways:

- (1) use the **bottom-up** splay method, which performs a standard BST search for e and then splay e to the root.
- (2) use the **top-down** splay method, which performs splaying while descending the search path.

(True/False) The resulting trees from these two methods will be identical.

Ex 1) $\text{Find}(3)$



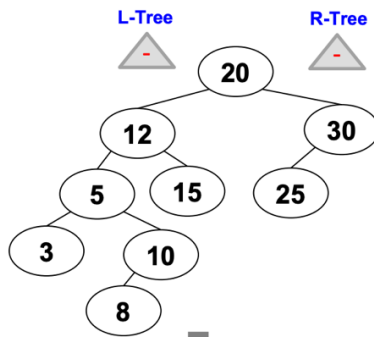
Ex 2) Find(0)

```

      5
     /\
    3  7
   /\
  1  4
 /\
0  2

```

4. Perform Insert(11) in top-down manner. When to stop and construct a new tree with root 11?



5. Perform delete 5 in top-down manner.

6. Compare time complexity:

Delete (e) vs Delete-top-down (e)