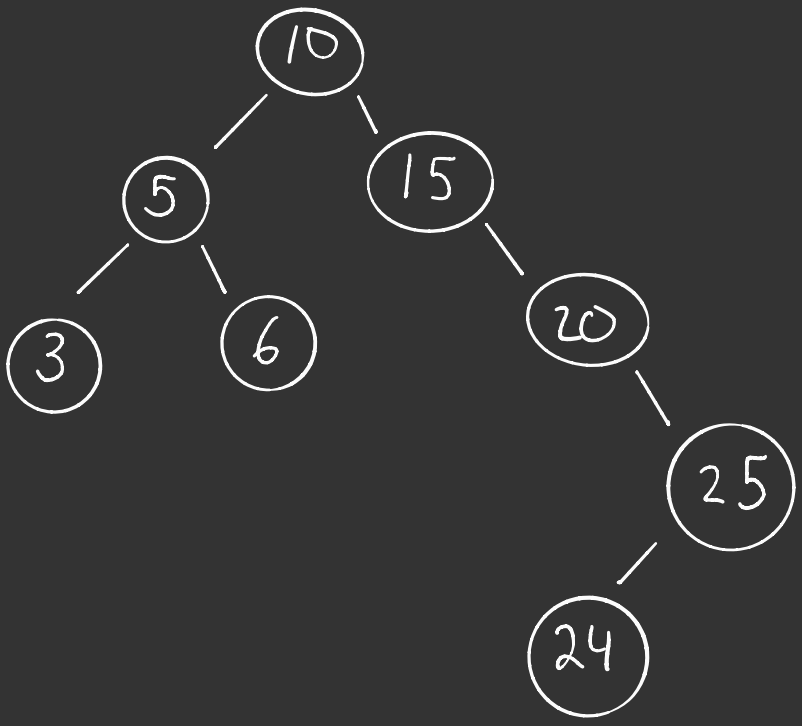
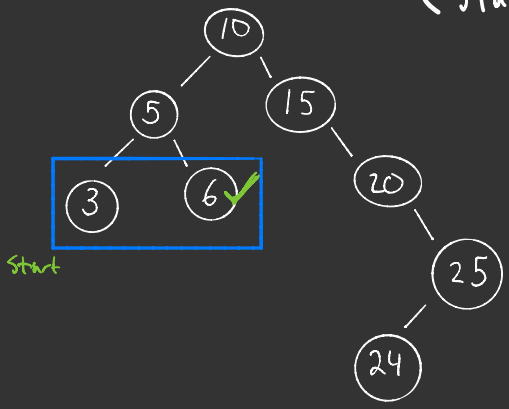
The asymptotic upper bound is O(n) for finding the top element (e.g. the item w/ the highest value in its key K) because the program must traverse through the BST constantly taking in the first largest K element, comparing it to the other elements, and replacing the largest K if an element that K compared to is greater than K.



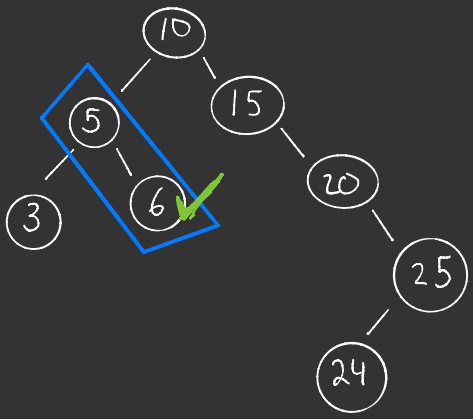
(start anywhere)



Compares 3 & 6,

6 is larger.

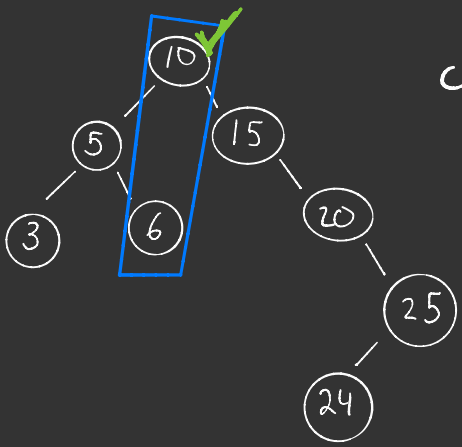
K = 6



Compares 6 & 5,

6 is larger.

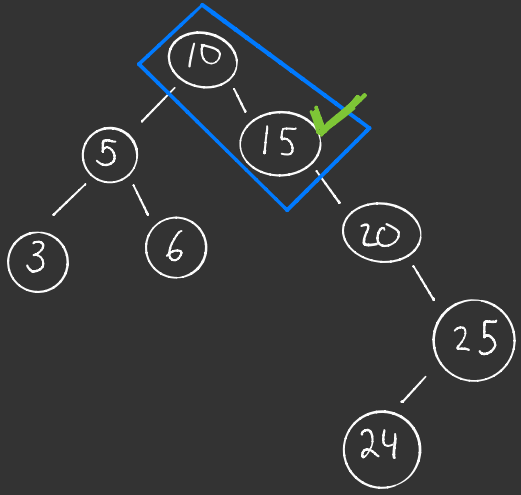
K = 6



Compares 6 & 10,

10 is larger.

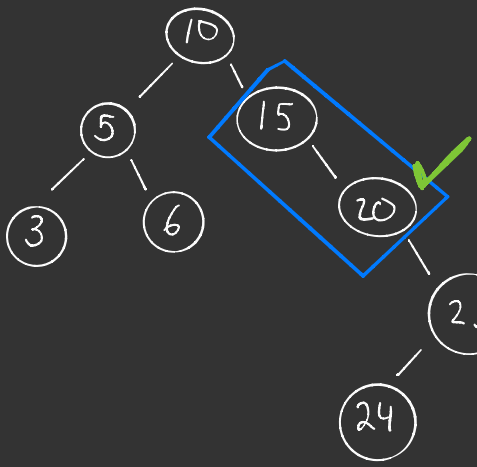
K = 10



Compare 10 & 15,

15 is larger

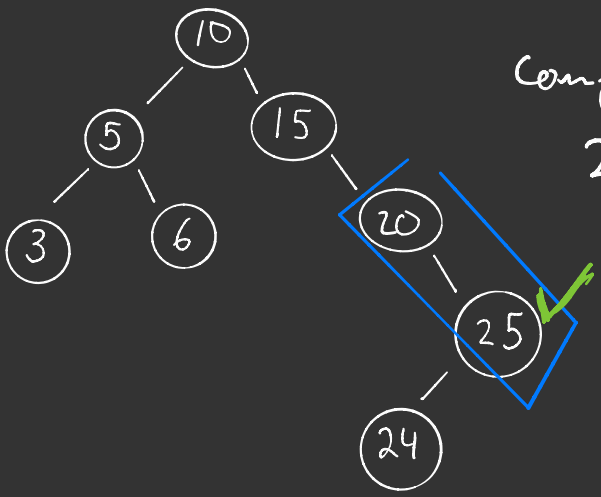
K = 15



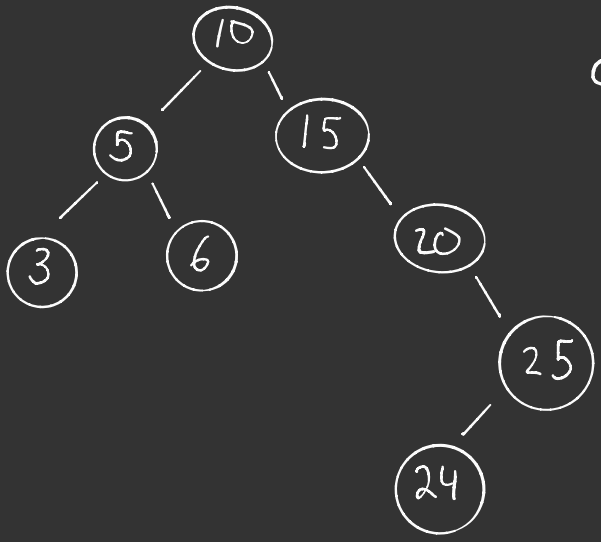
Compare 15 & 20,

20 is larger.

K = 20



Compare 20 & 25,
25 is larger.
K = 25



Compare 25 &
24, 25 is still
larger.
K = 25

so the largest element is 25.

From traversing through and comparing all elements ($O(n)$) and replacing k when element $>= k$, we found the k largest element. Hence, why the upper bound is $O(n)$.