- Given the root of a BST and a integer K return the Kth smallest element in this BST

→ To brute force it would map the whole BST values to an array, order it ascend. and then get the Kth element

Need to be smarter than that

- Need a global list var to store values

- lower values are on the left side

→ go all the way to the left first and store values in global var
  - as you go check list size with k
    - if it matches, stop and return the last element in the list

→ if you go all the left nodes and k doesn't match list size, start traversing right side nodes from

L> bottom to top

→ at some point left size will match K and you will have the answer

DFS algo applied as you will traverse wide left side first

) You always have to traverse left tree, because lower values

are at the bottom, but that is not necessarily true to right side

- So you go traversing right side go checking to see if K hasn't already been found by (list.size == k)
  - if that is true then quit.