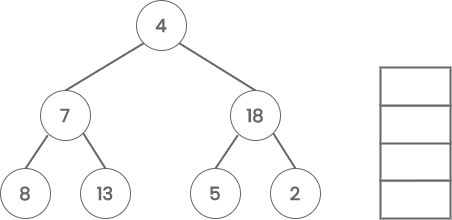
**In-order Traversal Without Recursion**

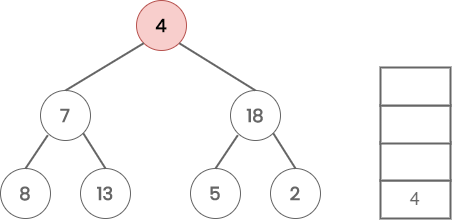
The following operations are performed to traverse a binary tree in in-order using a stack:

1. Start from the root, call it PTR.
2. Push PTR onto stack if PTR is not NULL.
3. Move to left of PTR and repeat step 2.
4. If PTR is NULL and stack is not empty, then Pop element from stack and set as PTR.
5. Process PTR and move to right of PTR , go to step 2.

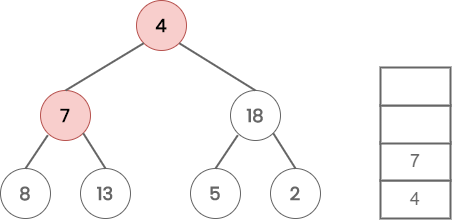
Consider the following tree.



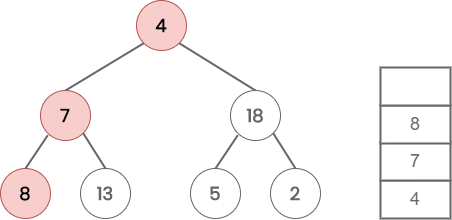
Start with node 4 and call it PTR. Since PTR is not NULL, PUSH it onto the stack.



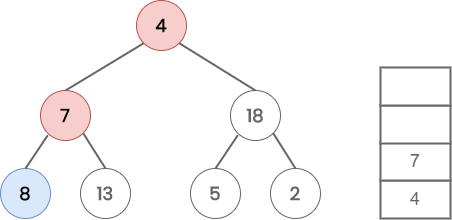
Move to the left of node 4. Now PTR is node 7, which is not NULL. So PUSH it onto the stack.



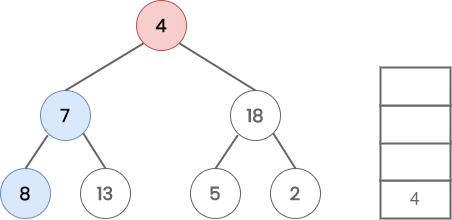
Again, move to the left of node 7. Now PTR is node 8, which is not NULL. So PUSH it onto the stack.



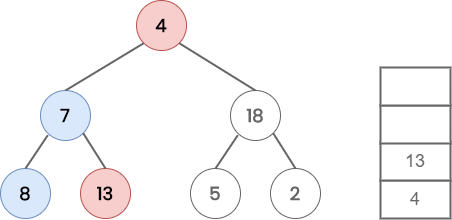
When we move again to the left of node 8, PTR becomes NULL. So POP 8 from the stack. Process PTR (8).



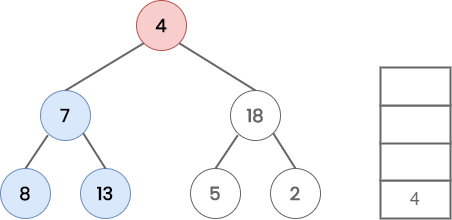
Move to the right child of PTR(8), which is NULL. So POP 7 from the stack and process it. Now PTR points to 7.



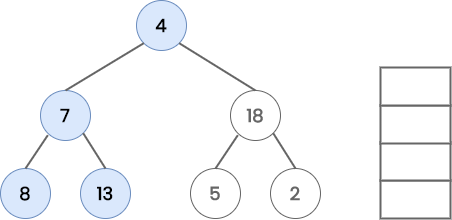
Move to the right child(13) of PTR and PUSH it onto the stack.



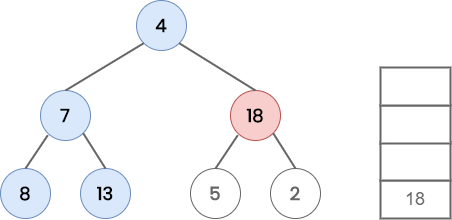
Move to the left of node 13, which is NULL. So POP 13 from the stack and process it.



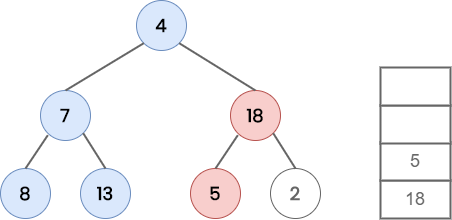
Since node 13 don’t have any right child, POP 4 from the stack and process it. Now PTR points to node 4.



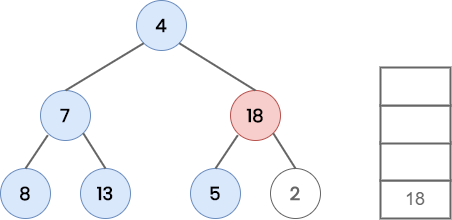
Move to the right of node 4 and put it on to the stack. Now PTR points to node 18.



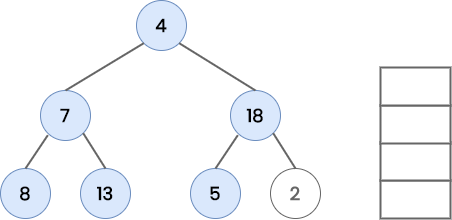
Move to the left(5) child of 18 and put it onto the stack. Now PTR points to node 5.



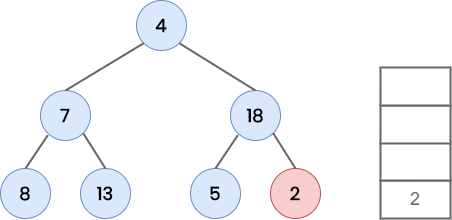
Move to the left of node 5, which is NULL. So POP 5 from the stack and process it.



Now, move to the right of node 5, which is NULL. So POP 18 from the stack and process it.



Move to the right(2) of node 18 and PUSH it on to the stack.



Since node 2 has left child, POP 2 from the stack and process it. Now the stack is empty and node 2 has no right child. So stop traversing.

