Inlustro Practical Examinations for DSA

Sharmila R
211082
3rd CSE B.

1.Finding intersecting node if we merge two singly linked list.

1.1.Algorithm for finding the merging point:

Step 1:First we need create nodes for singly linked list.

Step 2:And then create method for getIntersectionNode().

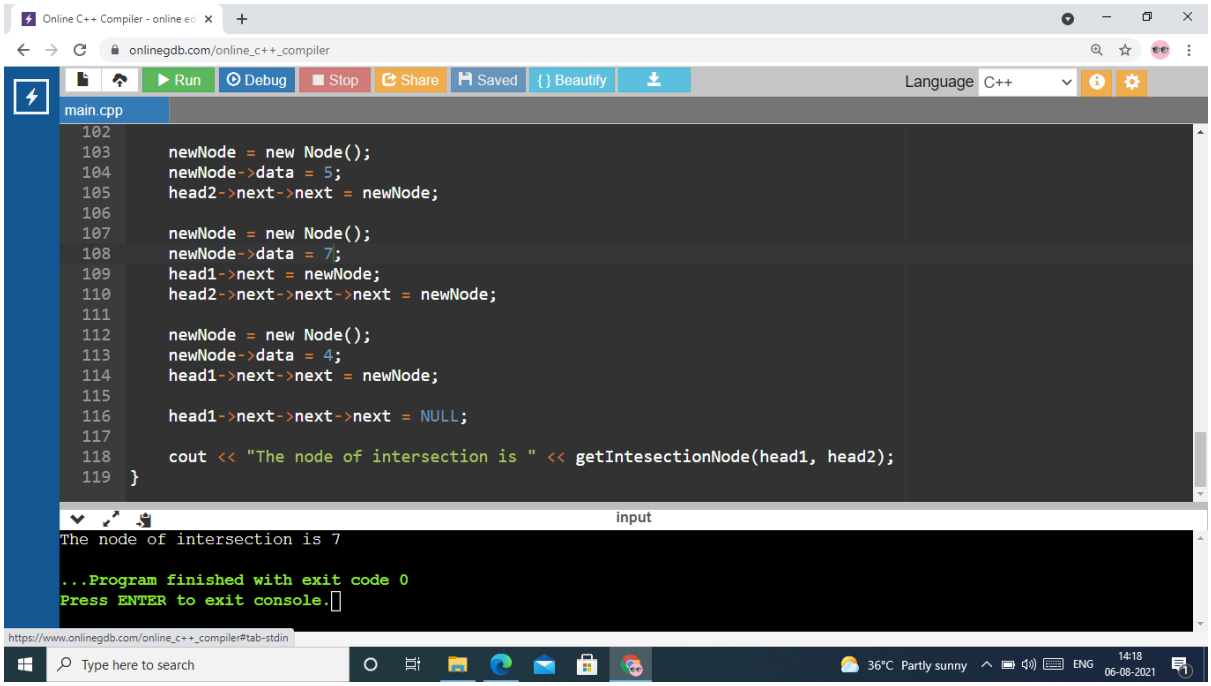
Step 3:Get count of the nodes in the first and second list,let count them as c1 and c2.

Step 4:Get the difference of counts if(c1>c2),then d=(c1-c2);else d=(c2-c1).

Step-5:Now traverse the bigger list from the lists equal no.of.nodes that from here onwards both the lists have equal no.of.nodes.

Step 6:Then we have to traverse both the lists in parallel till we come across a common node.

Step 7:Then we can find the merging point of the 2 linked lists.



1.2.Can we solve this using the sorting technique?

Yes,we can solve this problem using the sorting technique.We have to first create an array and store all the address of the nodes in this array.And then sort the array.For each element in the second list,we have to search fpr the address in the array.If we find a same memory address,then that is the merging point of the 2 points.

3. Can we solve it using hash tables?

Yes,we can solve this using problem using hash tables.

1.4 Can we use stacks for solving?

Yes,we can solve this using problem using stacks.We have to create 2 differeent stacks and push the elements in respective stacks.We have to pop the elements from the stack at once,till both lists are merged(we will getthe same value from both stacks).If both stacks returns different values then the last popped element is the merging point of the list.

1.5. Is there any other way of solving this ?

Yes,we can solve this problem sing other methods too.

Example:Brute-Force method.

1.6. Can we improve the complexity for?

Yes,we can improve the complexity for this problem.

2.Give an algorithm for finding the size of a binary tree.

Step 1: 1.Size() recursively calculates the size of a tree.

Step 2:The size of the tree is calculated by using the formula: Size of a tree=Size of left subtree+1+Size of right suubtree.

Step 3:call the method Size().

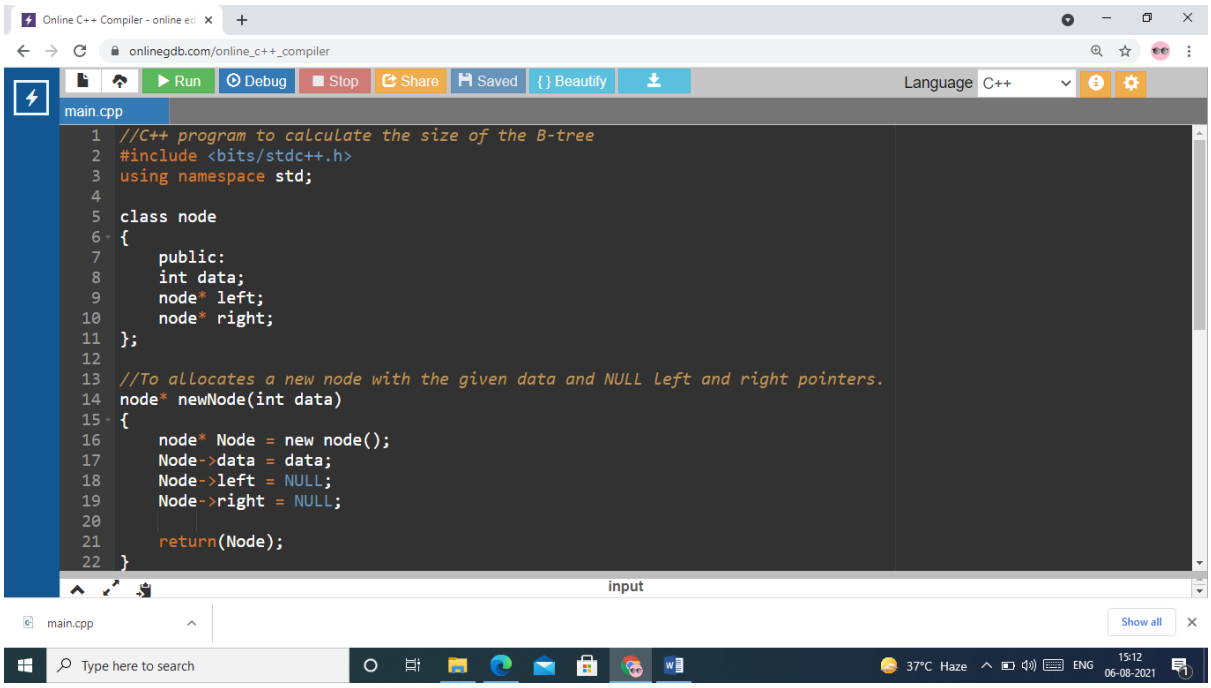
i)If tree is empty it will return 0.

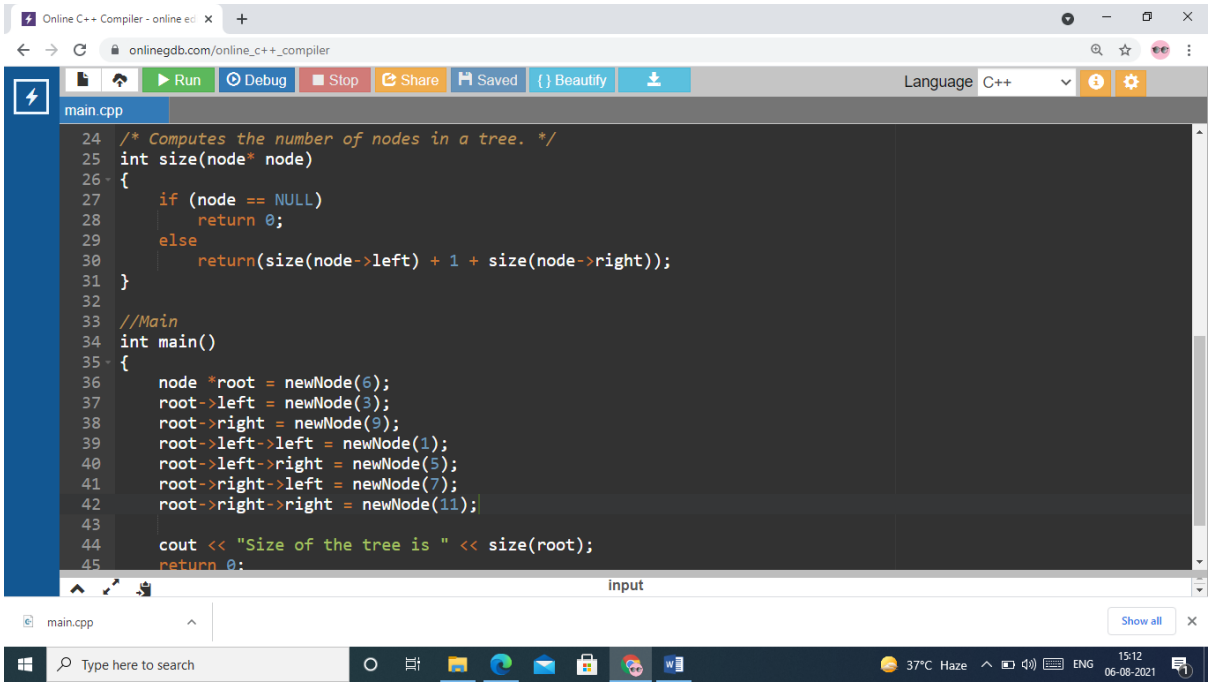
ii)Else it get the size of left and right subtree recursively.

Then calculate the size of the tree.

Step 4:Then we have to give input for B-tree.

Step 5:Then it will print the size of B-tree.





2.a) How will you solve it without recursion?

If we don’t use recursive,we need a data structure to store the tree traversal,we will use queue .