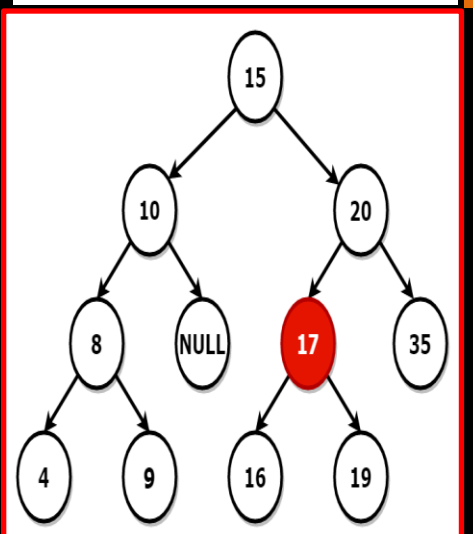
Case3: Delete a child with two children

Stack 1
main()
delete(root,17)

Stack 2
delete(ptr, 17) ->ptr is 15
if(ptr==NULL) -> False
else if(17<15) -> False
else if(17>15) -> True
ptr->right = delete(ptr->right,17)
15->right becomes 20
return node 15 to main function

Input tree



Stack 3
delete(ptr, 17) -> ptr is 20
if(ptr==NULL) ->False
else if(17<20) -> True
ptr->left = delete(ptr->left,17)
20->right becomes 35
return the node 20 to prev recursive call

Stack 4
delete(ptr, 17) -> ptr is 17
if(ptr==NULL) ->False
else if(17<17) -> False
else if(17>17) -> False
else -> True
if(ptr->left &&ptr->right) ->True
temp = findNodeWithMinVal(ptr->right)
returns 19 from RST of node 17
ptr->data = temp->data = 19
ptr->right = delete(ptr->right, 19)
return ptr as node17 to prev recursive call

Stack 5
delete(ptr, 19) -> ptr is 19
if(ptr==NULL) ->False
else if(19<19) -> False
else if(19>19) -> False
else -> True
if(ptr->left &&ptr->right) ->False
else -> temp = ptr
if(ptr->left == NULL) ->True
ptr = ptr->right
free(temp) -> delete the node 19
return ptr as NULL to prev recursive call

Output Tree