## Deleting a node

Find the node to be deleted. Once found then one of the following conditions will apply:

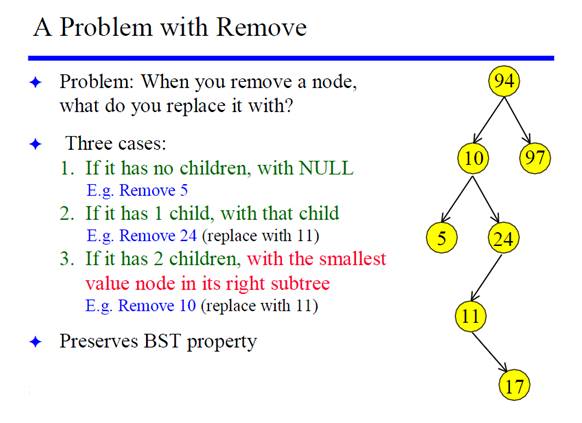
1. The node being deleted has no children (leaf node)
   * Delete, connect the null to its parent
2. The node being deleted has one child (either on the right or the left)
   * Take either the left or the right child and point it to the parent of the existing node that is to be deleted.
   * Delete the existing node.
3. The node being deleted has two children

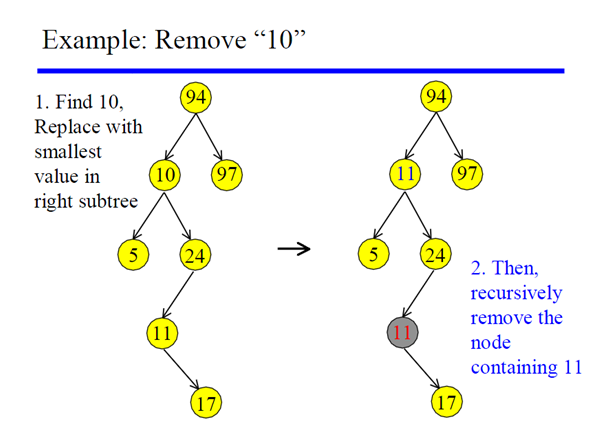
Two options

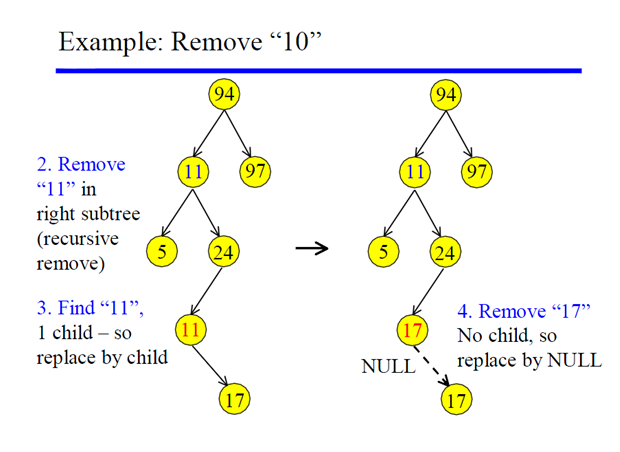
* + 1st option
    - Go to the right child. Then go continuously to the left until you get to the leaf node(Smallest in the subtree after the right child)
  + 2nd option
    - Go the left child. Then go continuously to the right until you get to the leaf node (Largest in the subtree after the left child)

Take the value of the leaf node from option 1 or option 2 and substitute that value with the value of the node that is to be deleted.

Delete the leaf node that was identified in option 1 or option 2.







7

3---------------20

2-------6 9 -----------23

22------X

21------X

Delete 2 (If no child)

|  |  |
| --- | --- |
| Going | Coming back |
| Del (7, 2)  leaf->left=del(3, 2) | leaf -> left =3 //connect 3 to 7  Return 7 |
| Del (3, 2)  leaf -> left =del(2, 2) | leaf ->left=null //connect null to 3  Return 3 |
| Del (2, 2)  Temp=null  Delete 2  leaf =null (Temp)  Return null (leaf) |  |

7

3---------------20

2-------6 9 -----------23

22------X

21------X

Delete 23 (If one child)

|  |  |
| --- | --- |
| Going | Coming back |
| Del (7, 23)  leaf ->right=del(20, 23) | leaf ->right=20 //connect 7 to 20  Return 7 |
| Del (20, 23)  leaf ->right=del(23, 23) | leaf ->right=22 //connect 20 to 23  Return 20 |
| Del (23, 23)  Temp=22  Delete 23  leaf =22 (temp)  Return 22 (leaf) |  |

7

3---------------20

2-------6 9 -----------23

22------X

21------X

Delete 20 (If two children)

|  |  |
| --- | --- |
| Going | Coming back |
| Del (7, 20)  leaf ->right=del(20, 20)  Return leaf | leaf ->right=21 //connect 7 to 21  Return 7 |
| Del (20, 20) //20 becomes 21  Temp=MIN(23)  leaf ->value=21 (temp)//sub 20 with 21  leaf ->right=Del(23, 21)  Return leaf | leaf ->right=23 //connect 21 to 23  Return 21 |
| Del (23, 21)  leaf ->left=del(22, 21)  Return leaf | leaf ->left=22 //connect 23 to 22  Return 23 |
| Del (22, 21)  leaf ->left=del(21, 21)  Return leaf | leaf ->left=NULL //Connect 22 to null  Return 22 |
| Del (21, 21) (No children)  Temp=null  Delete 21  leaf =null (temp)  Return null (leaf) |  |

23

22------X

21------X

Delete 20, min function (If two children)

|  |  |
| --- | --- |
| Going | Coming back |
| MIN(23)  current = MIN (22)L  return current | current=21  return 21 |
| MIN (22)  current = MIN (21)L  return current | current=21  return 21 |
| MIN (21)  return 21 |  |