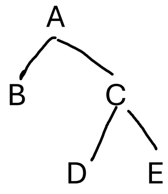
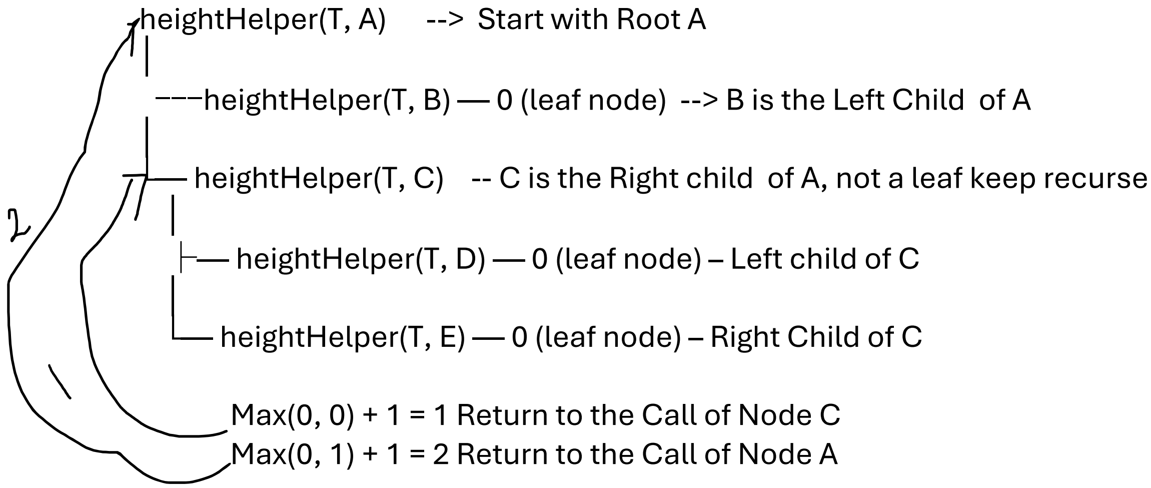
Finding the height of the Binary Tree

```
Algorithm height(T)
        return heightHelper(T, T.root()) // Start by passing a Tree T and Root

Algorithm heightHelper(T, p)
        if T.isExternal(p) then // Return 0 if p is leaf
                return 0
        // Recursively find the Left Subtree Height
        lheight := heightHelper(T, T.leftChild(p))

       // Recursively find the Right Subtree Height
        rheight := heightHelper(T, T.rightChild(p))

        // Return the height which has Maximum
        return MAX(lheight, rheight) + 1 // +1 wilk make count of move from
                                         //   parent to child
```

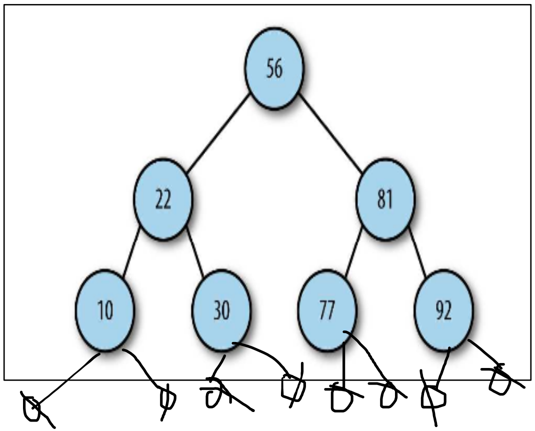
Example:





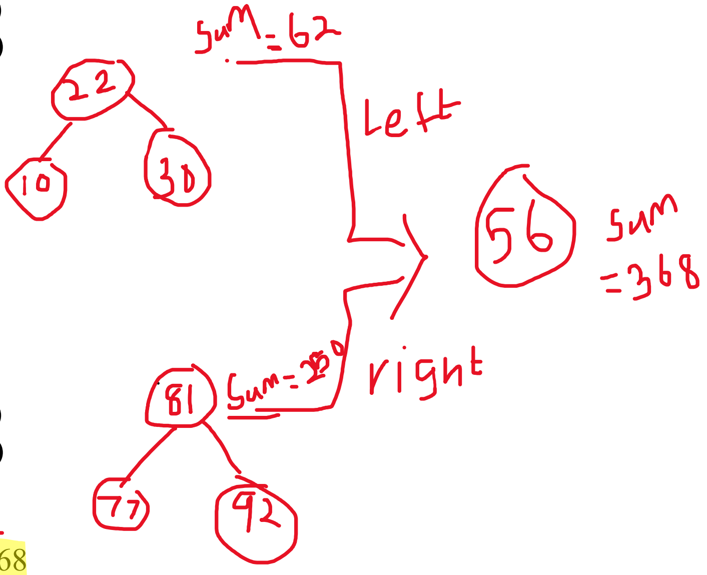
```
heightHelper(T, A)     -->  Start with Root A
 |
 ---heightHelper(T, B) — 0 (leaf node)  --> B is the Left Child  of A
 |
 ├── heightHelper(T, C)    -- C is the Right child  of A, not a leaf keep recurse
      |
      ├— heightHelper(T, D) — 0 (leaf node) – Left child of C
      |
      └— heightHelper(T, E) — 0 (leaf node) – Right Child of C

          Max(0, 0) + 1 = 1 Return to the Call of Node C
          Max(0, 1) + 1 = 2 Return to the Call of Node A
```

```
Algorithm sum(T)
        return sumHelper(T, T.root())

Algorithm sumHelper(T, p)
// Condition to check the P is null
        if T.isExternal(p) then
                return 0
        lsum := sumHelper(T, T.leftChild(p))
        rsum .:= sumHelper(T, T.rightChild(p))
        return lsum + rsum + p.element()
```



// lsum is the sum of Lchild, rsum is the sum of rchild, p.element() is the parent value

```
|--- sumHelper(56)
   |--- sumHelper(22)
   |  |--- sumHelper(10)
        |  |  |--- 0 (base case for leaf stored at lsum)
        |  |. |--- 0 (base case for leaf stored at rsum)
                  ---- 0 + 0 + 10 = 10
   |  |--- sumHelper(30)
        |--- 0 (base case for leaf stored at lsum)
        |--- 0 (base case for leaf stored at rsum)
      ---- 0 + 0 + 30 = 30
   |  |--- 10 + 30 + 22 (sum at Node 22) = 62
   |
   |--- sumHelper(81)
   |  |--- sumHelper(77)
      |  |--- 0 (base case for leaf stored at lsum)
      |. |--- 0 (base case for leaf stored at rsum)
      ---- 0 + 0 + 77 = 77
   |  |--- sumHelper(92)
        |--- 0 (base case for leaf stored at lsum)
        |--- 0 (base case for leaf stored at rsum)
      ---- 0 + 0 + 92 = 92
   |  |--- 77 + 92 + 81 (sum at Node 81) = 250
   | 62 + 250 + 56 = 368 ( sum at Node 56) = 368
```


sum=62
Left
22
10
30
56
sum
=368
81
Sum=250
right
77
92