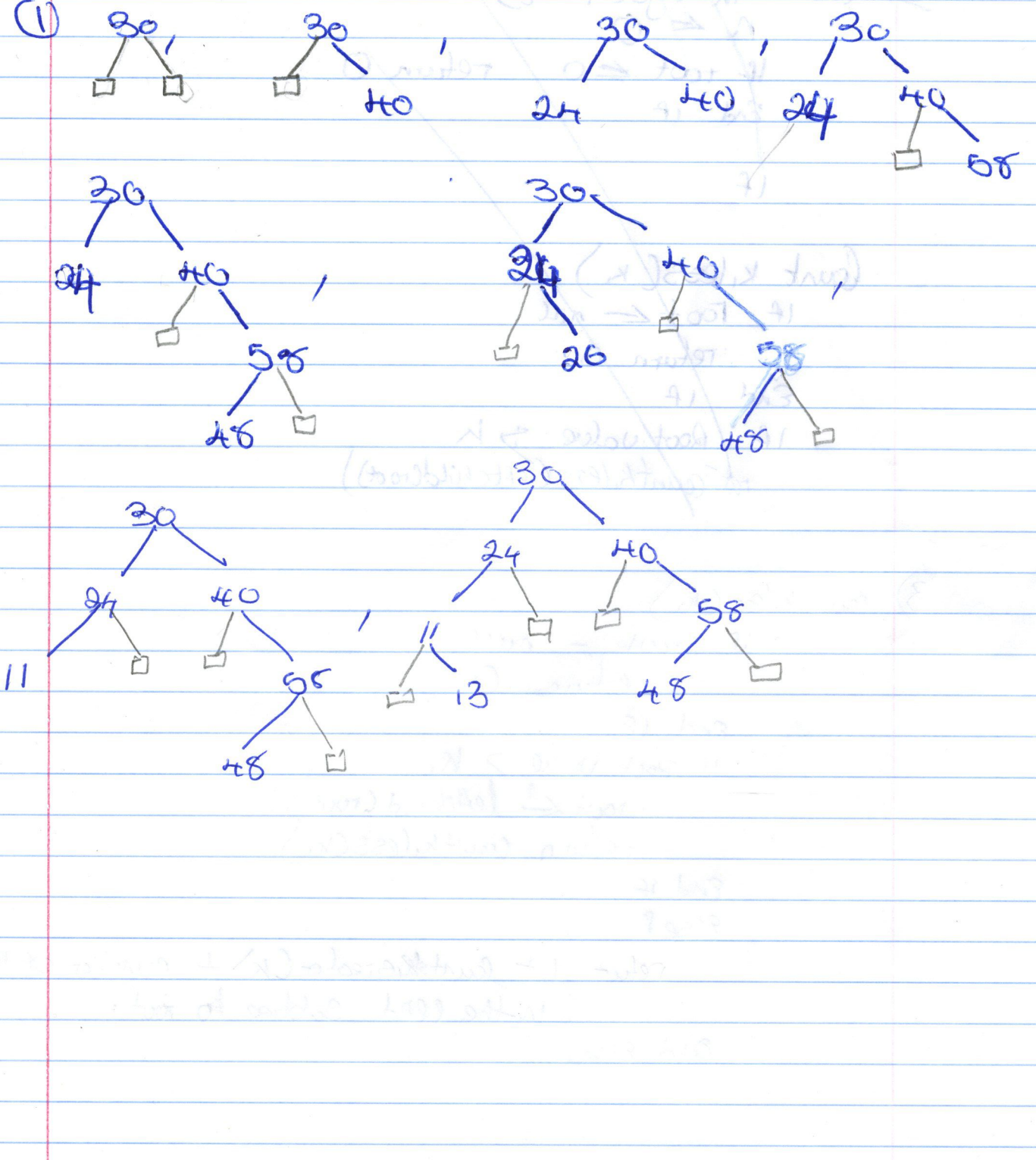
2.4 * 32.

# CS2 225 Assignment

(1)



3 countAllInRange(k1, k2)

n ← 0

If root ≤ 0 return 0

End if

If

countk1less(k1)

If root ← null

return 0

End if

If Root.value ≥ k

root = countk1less(leftchild(root))

3) countk1less(k1)

If root = null

return 0

End if

If Root value > k1

root ← leftchild(root)

return countk1less(k1)

End if

Else?

retur 1 + countk1greater(k) + number of nodes in the left subtree to root.

End Else.

Count $k_1$, Greater ($k_2$)

counts all element greater than k

same function as count $k_1$ (root) but instead of traversing to the left we traverse to the right

returns 1 + Count $k_1$, Greater ($k_2$) + all number of nodes in the right subtree.

CalculateAllNode ($k_1$, $k_2$)

```
If root = null
    return 0
End if
If (root value < k1)
    traverse through the right subtre recursively
    root = rightchild(root)
    CalculateAllNode(k1, k2)
End if
If (root.value > k1)
    traverse through the left subtree recursively
    root = leftchild(root)
    CalculateAllNode(k1, k2)
End if
else
    return 1 + Sum of All left tree + Sum of All
        right tree
End else
```