CS 5633: Analysis of Algorithms

Homework 5

1. func find\_median(A[1….n], B[1….n])

mid = Ln/2亅

if (sizeof(A) == 1) //both arrays will have same size in all steps

return(A[1]+B[1])/2

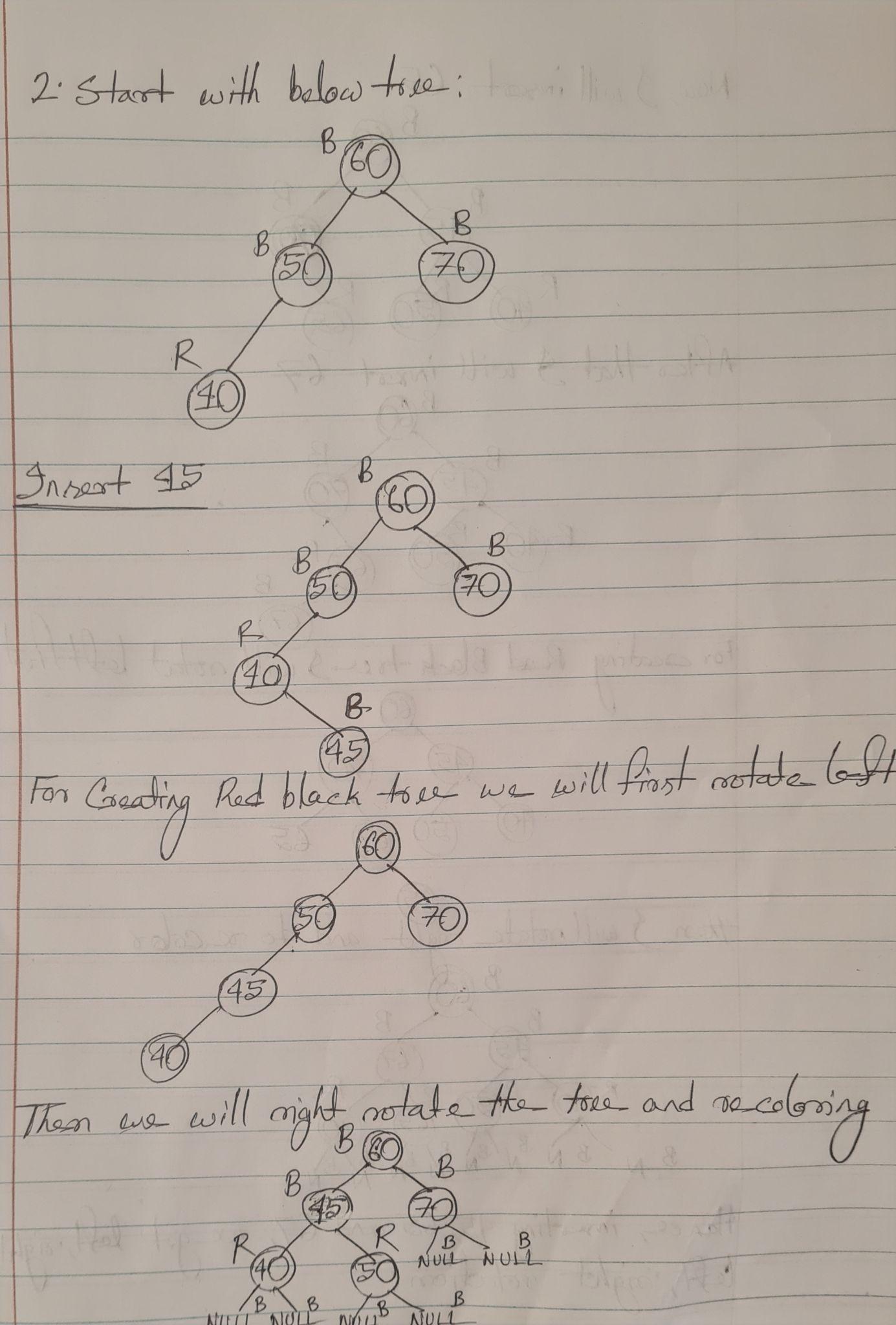
else if(A[mid] > B[mid])

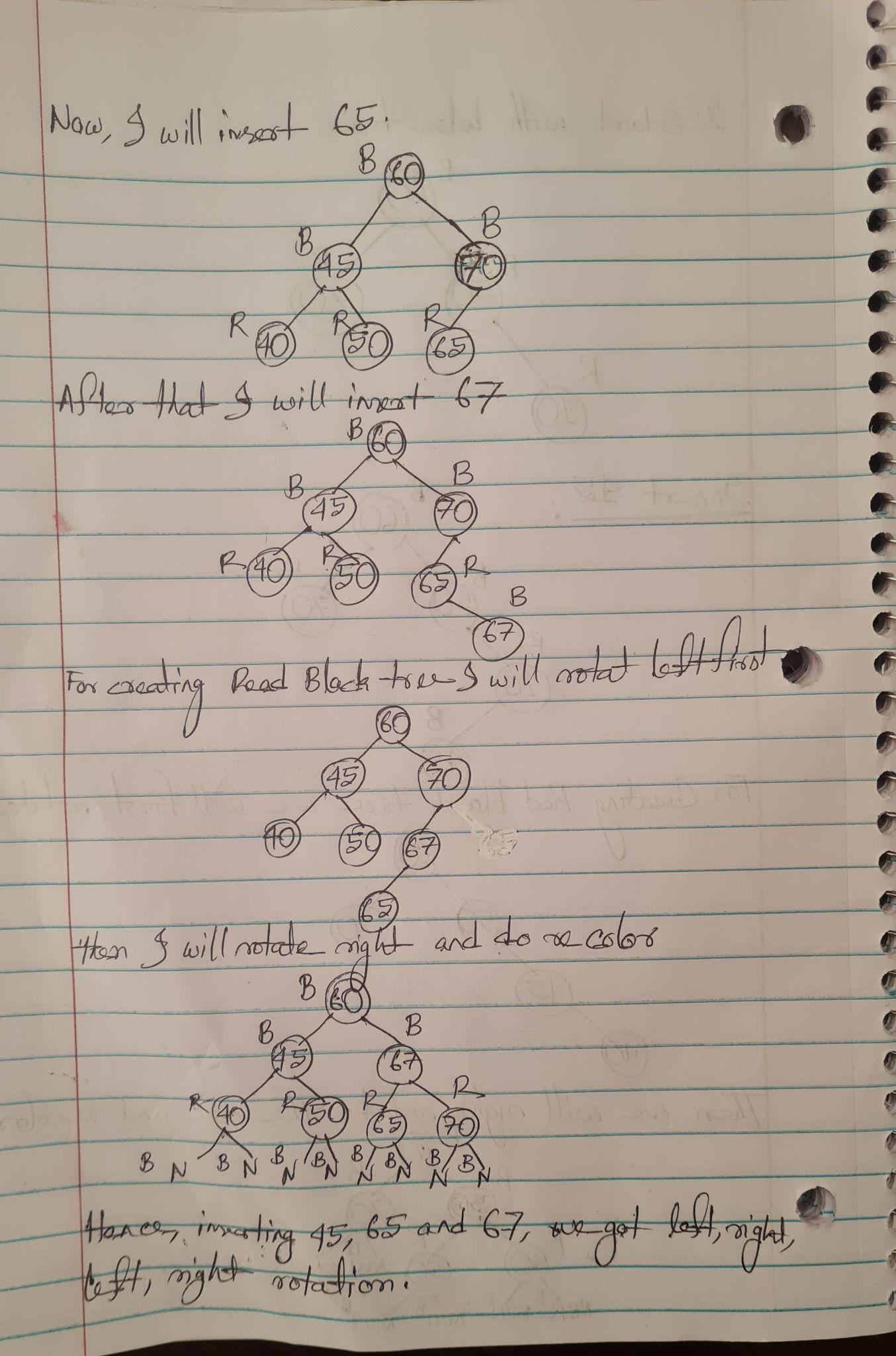
return find\_median(A[1….mid], B[mid+1…..n])

else

return find\_median(A[mid+1….n], B[1…...mid])

In each step, the problem size is reduced by half. So, the run time = O(lgn)

2.

3. a) A Red-Black tree will have maximum internal nodes if it has alternating red and black nodes and if it is a complete binary tree. So, the black height will be half of the actual height, b = h/2, => h = 2b. So, total internal nodes = 22b - 1.

b) To have the minimum number of internal nodes, the Red-Black tree will have no red nodes as if there is a red node there must have two black nodes. So, the actual height of the tree will be the black height, h = b. So, total internal nodes = 2b - 1