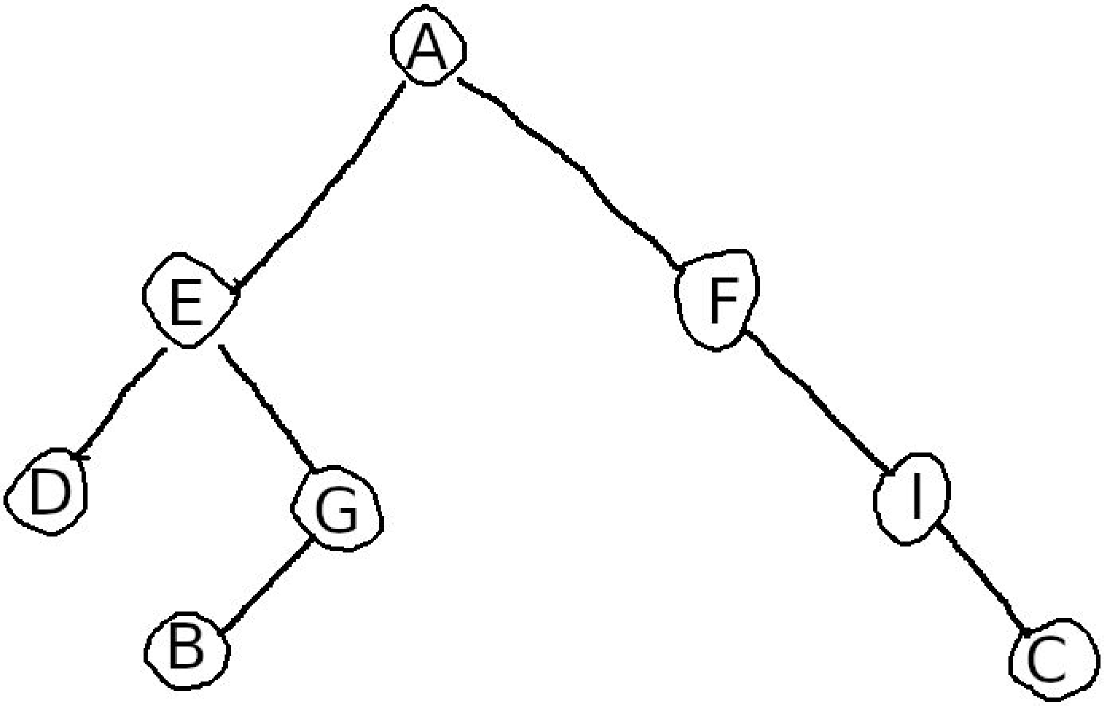
Gerald Hoff
Advanced Data Structures Java
HomeWork 2
10/20/2020

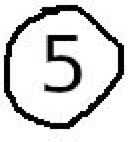
https://github.com/geraldHoff/Java---HW2

1)

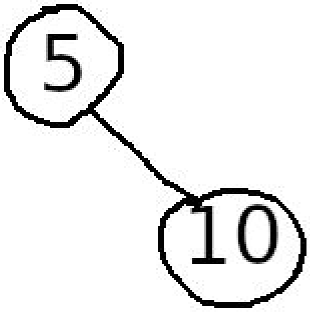


2)

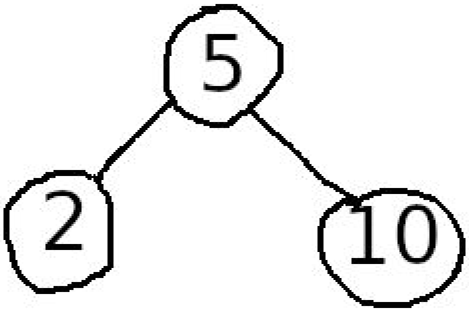
Step one:



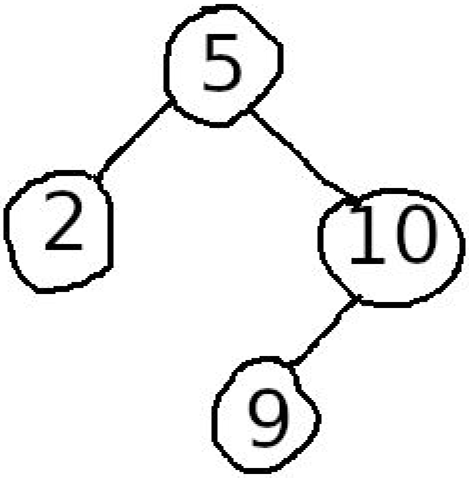
Step two:



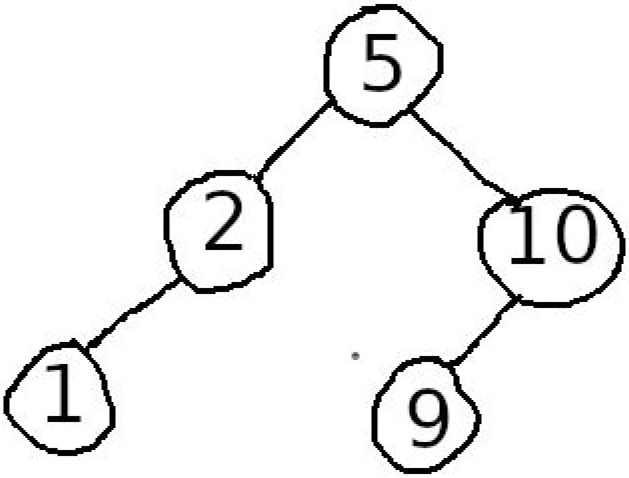
Step three:



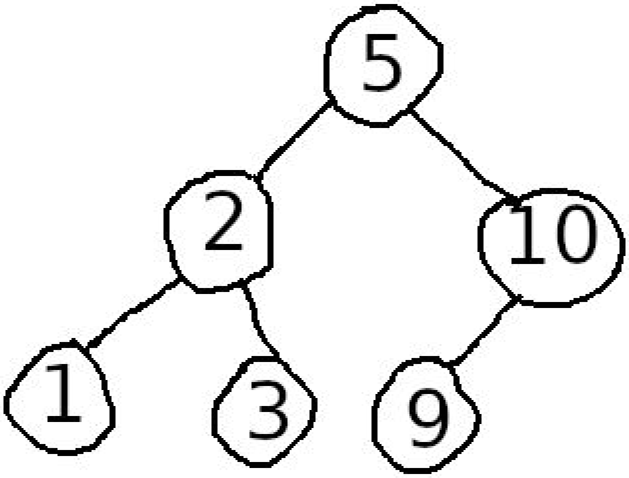
Step four:



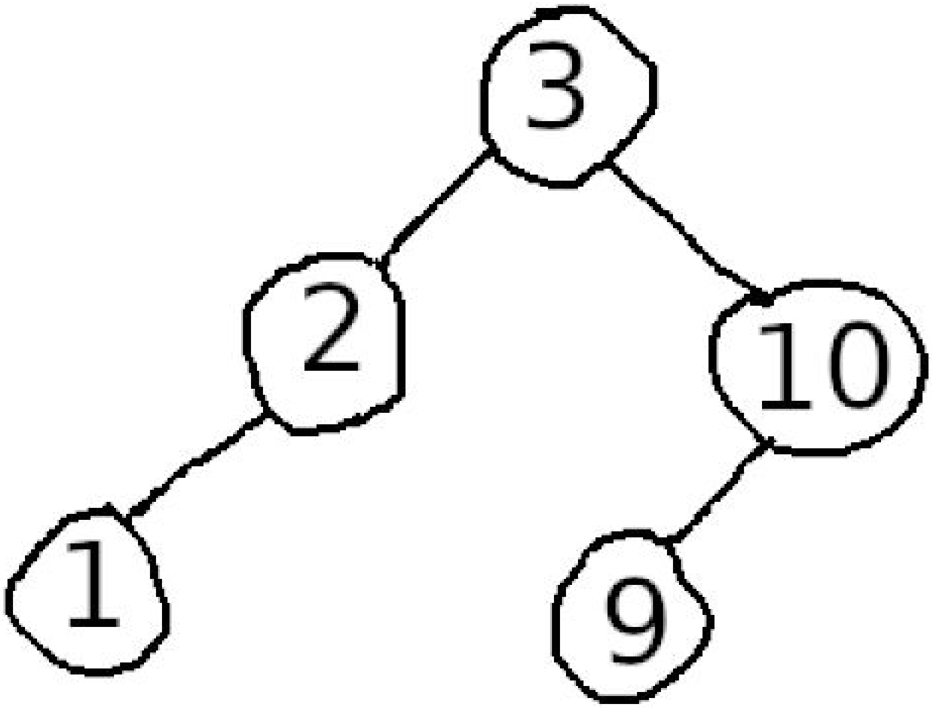
Step five:



Step six:

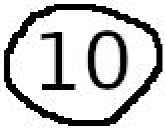


Step seven:

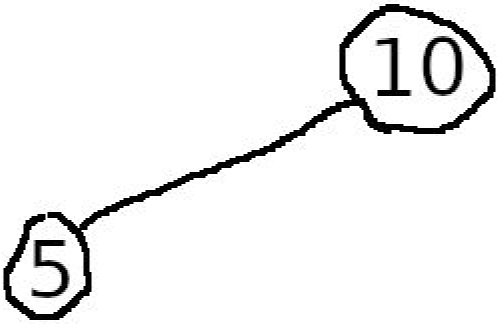


3)

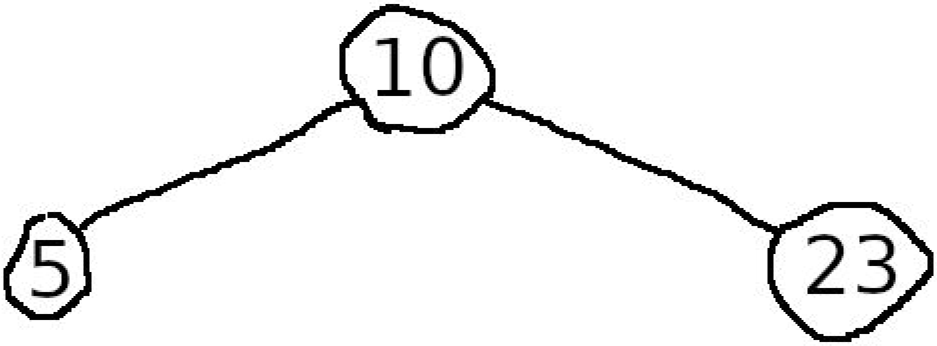
Step one:



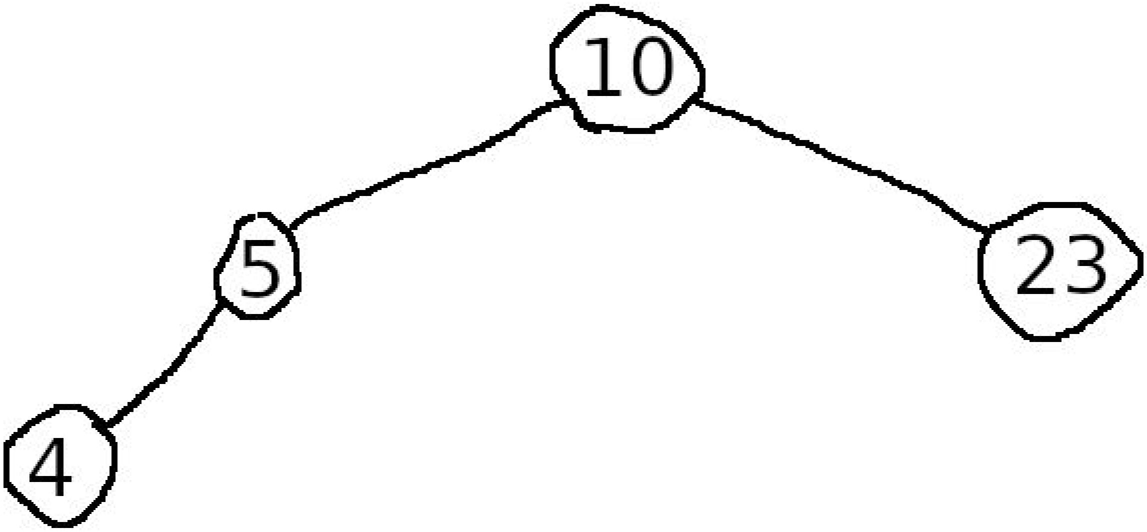
Step two:



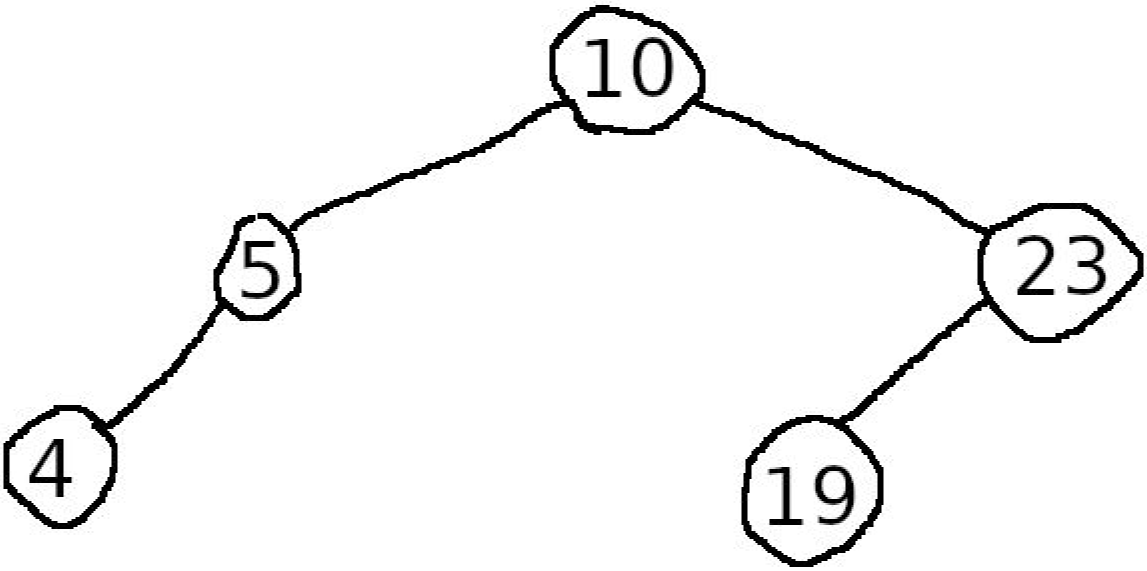
Step three:



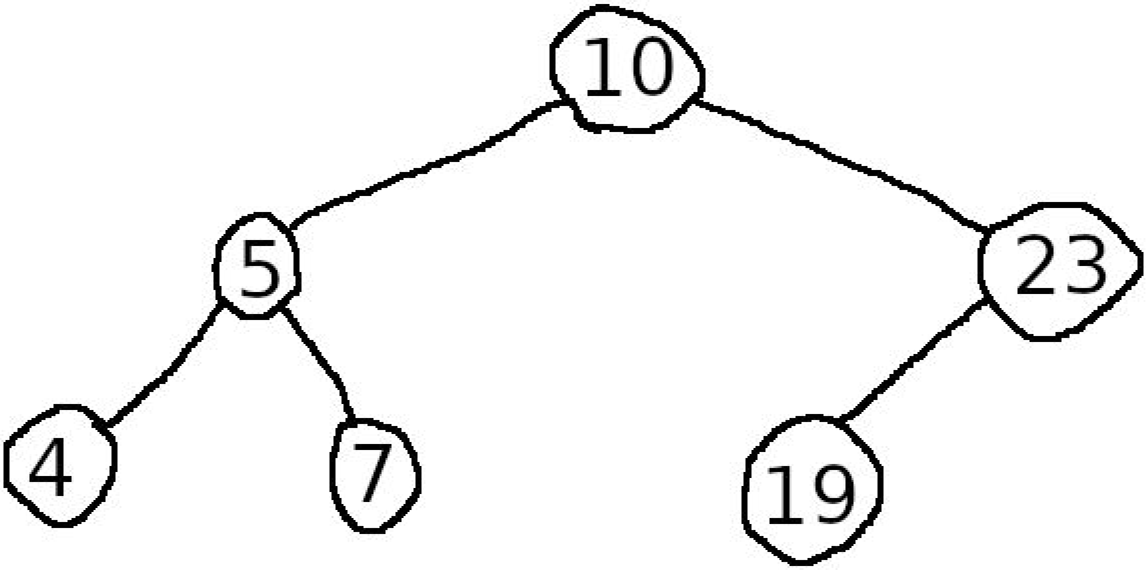
Step four:



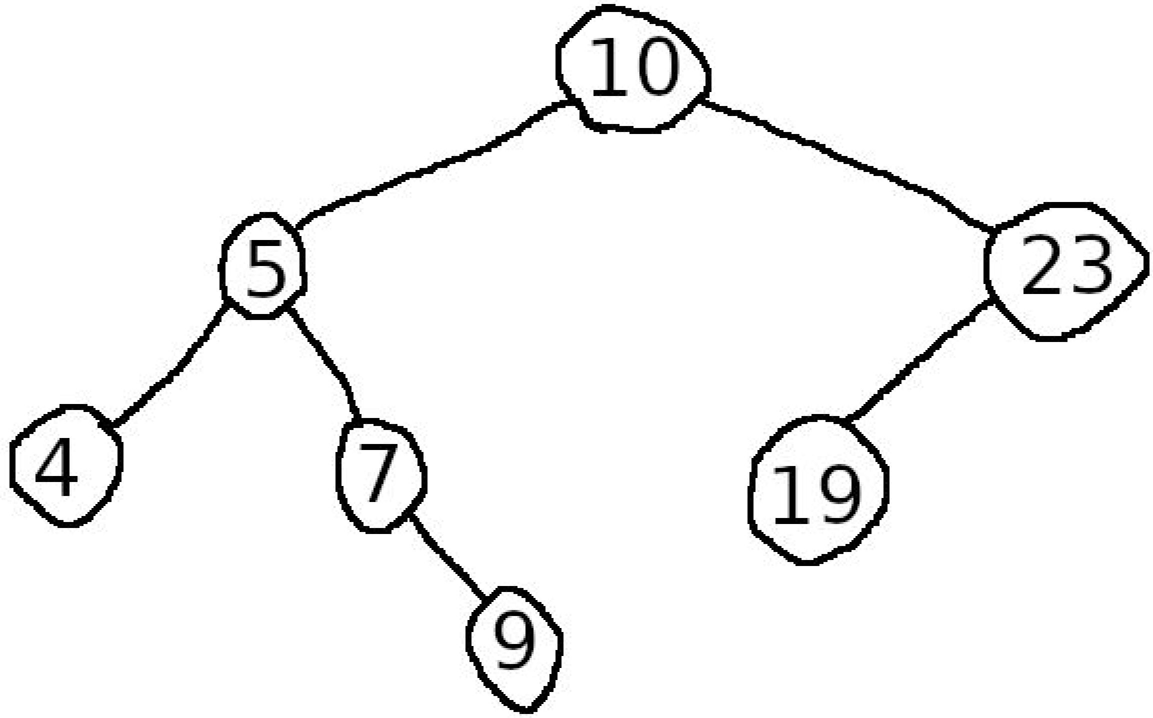
Step five:



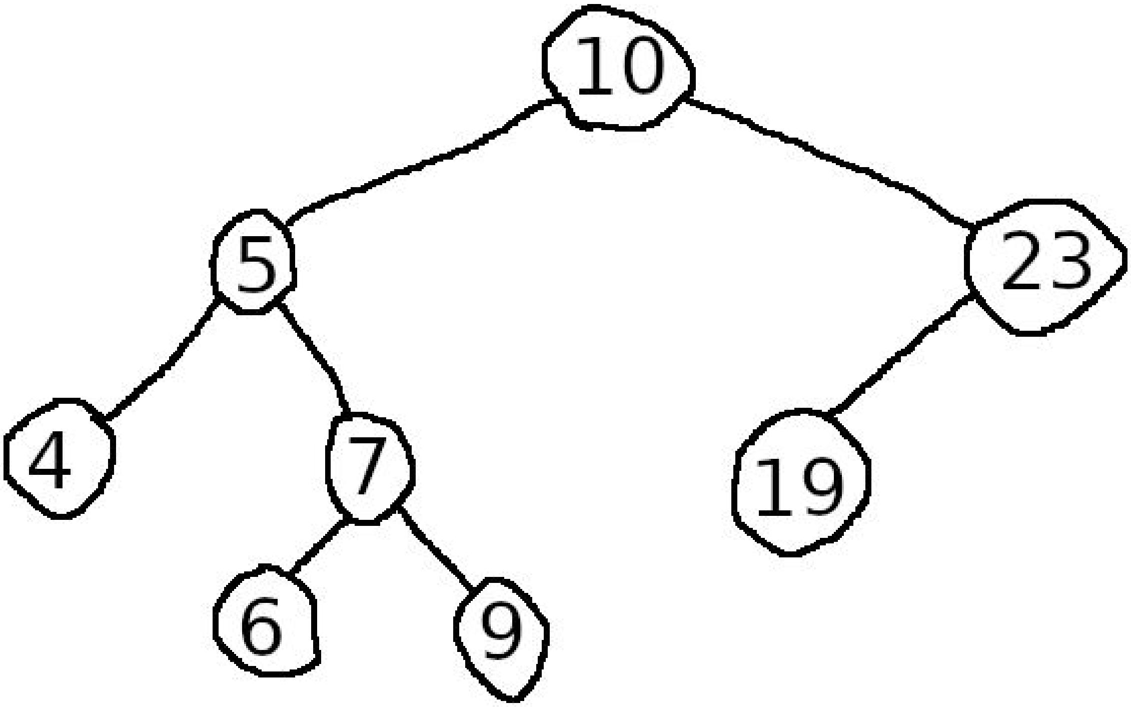
Step six:



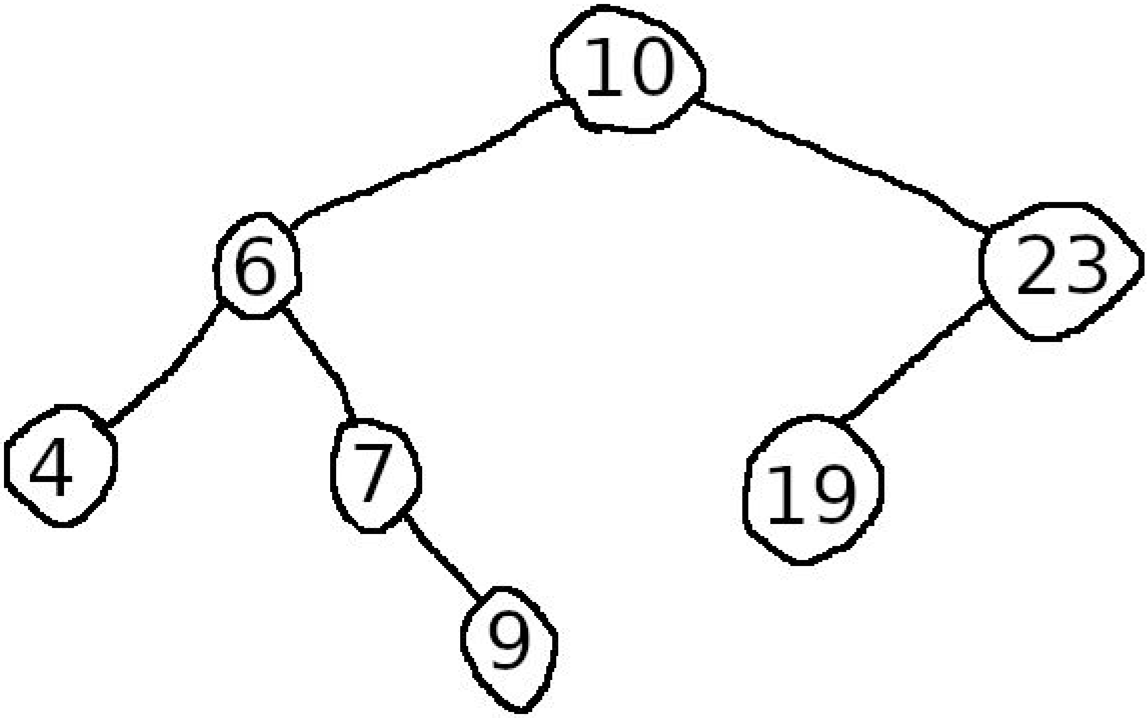
Step seven:



Step eight:



Step nine:



4)

a) 4

b) 3

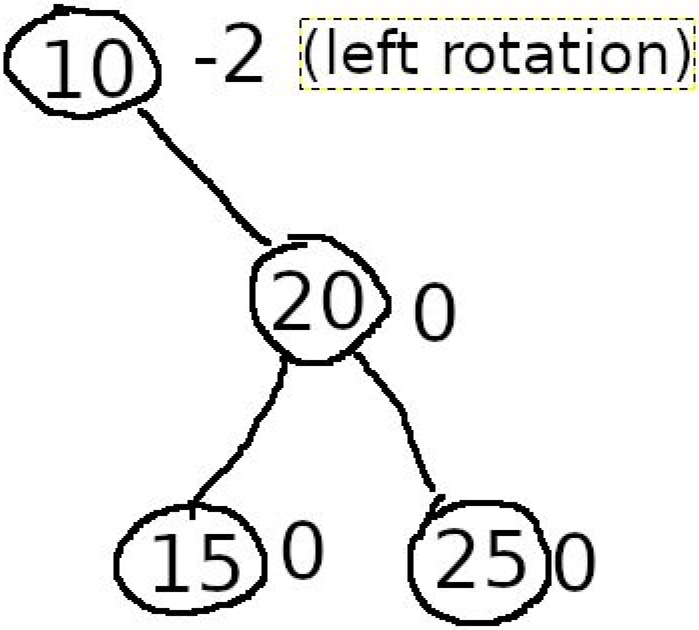
c) 1

d) Pre-order: 100, 50, 3, 1, 20, 80, 52, 90, 83, 99, 150, 125, 152.

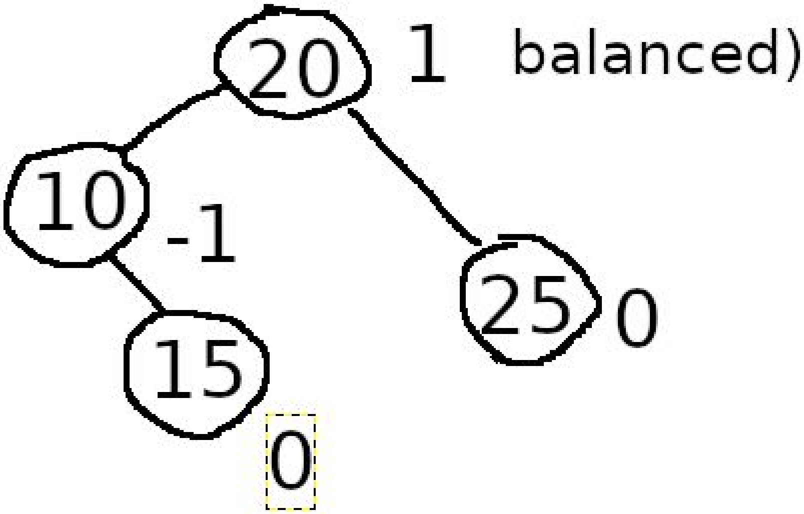
In-order: 1, 3, 20, 50, 52, 80, 83, 90, 99, 100, 125, 150, 152.

Post-order: 1, 20, 30, 52, 83, 99, 90, 80, 50, 125, 152, 150, 100.

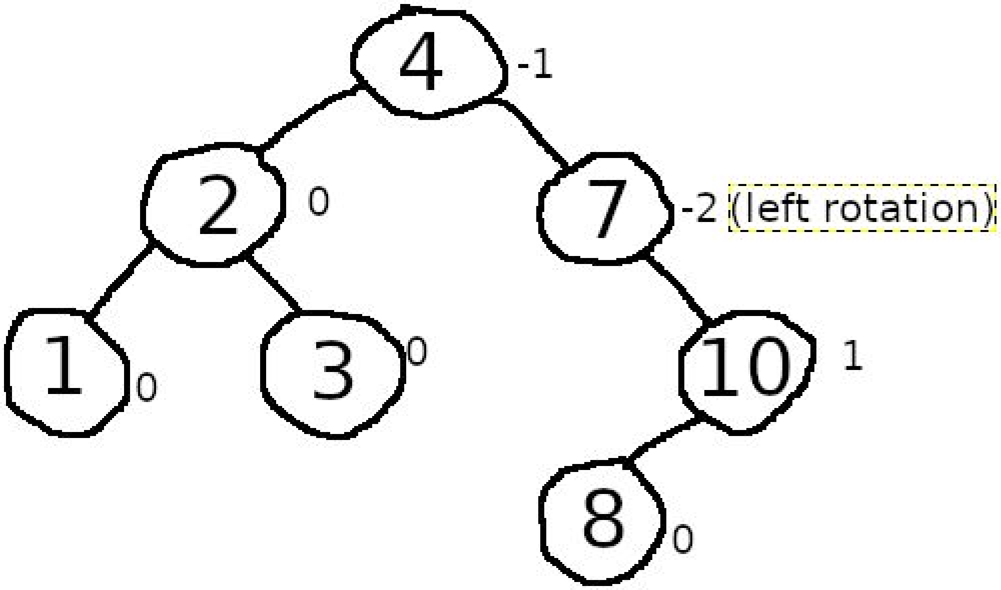
5)



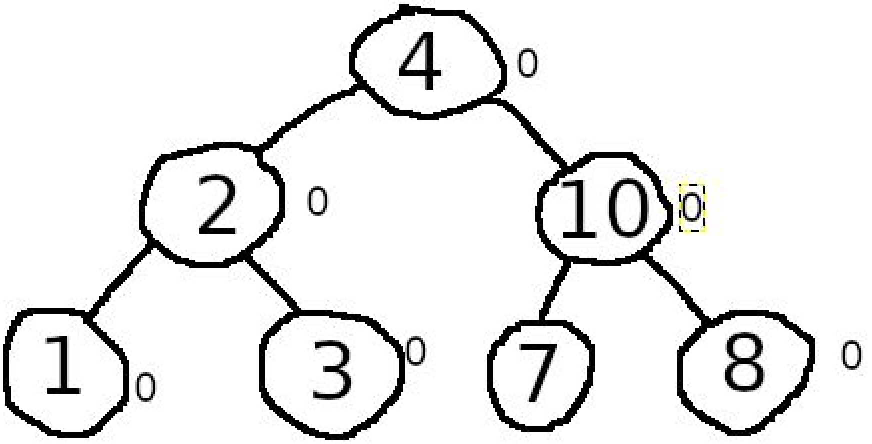
After rotation:



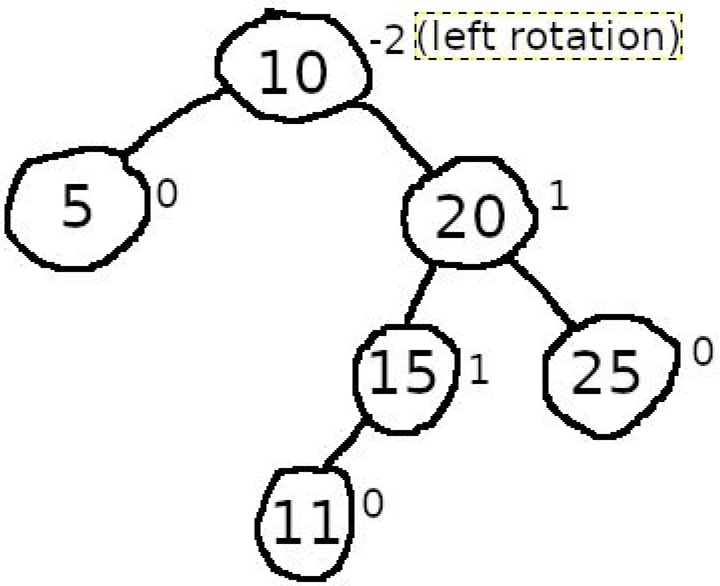
6)



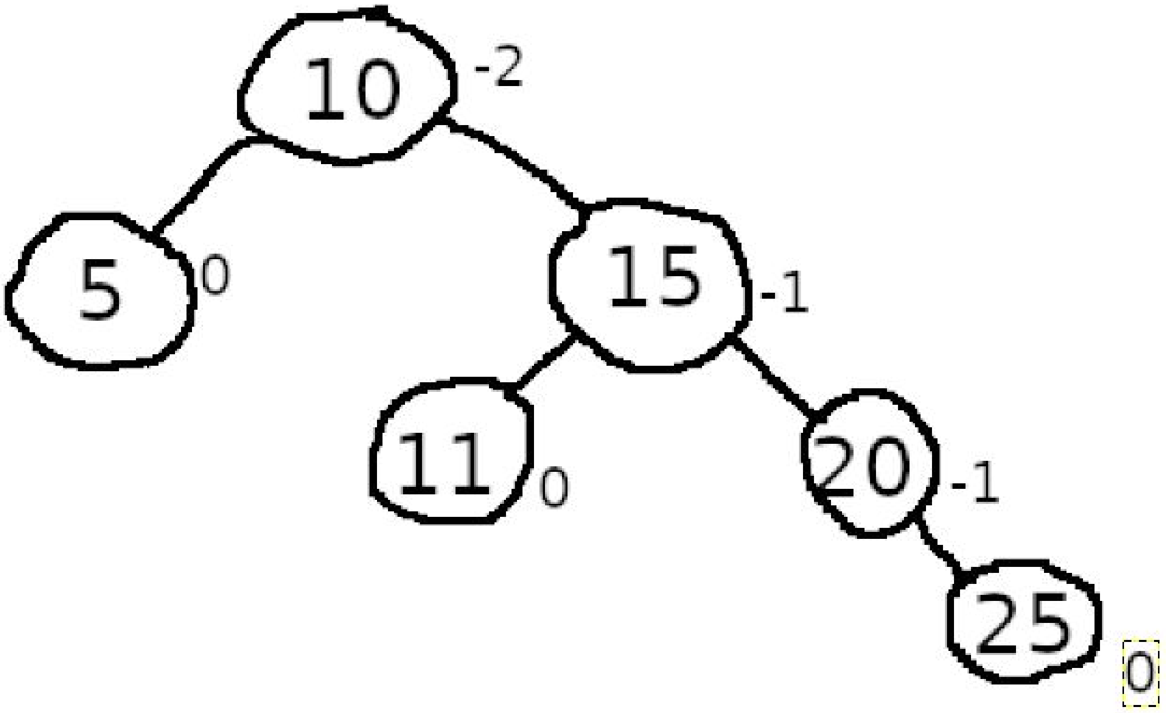
After rotation:



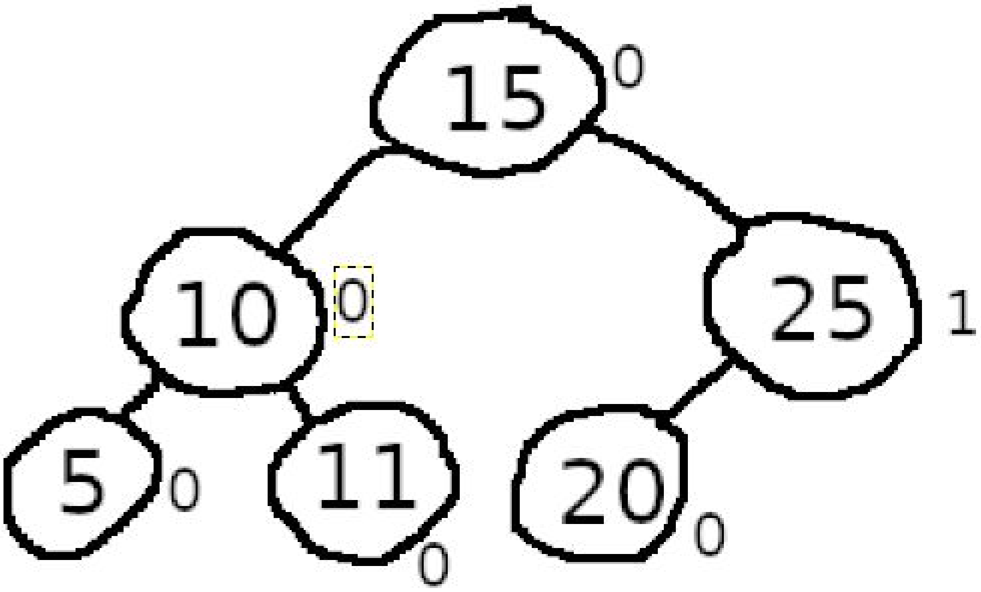
7)



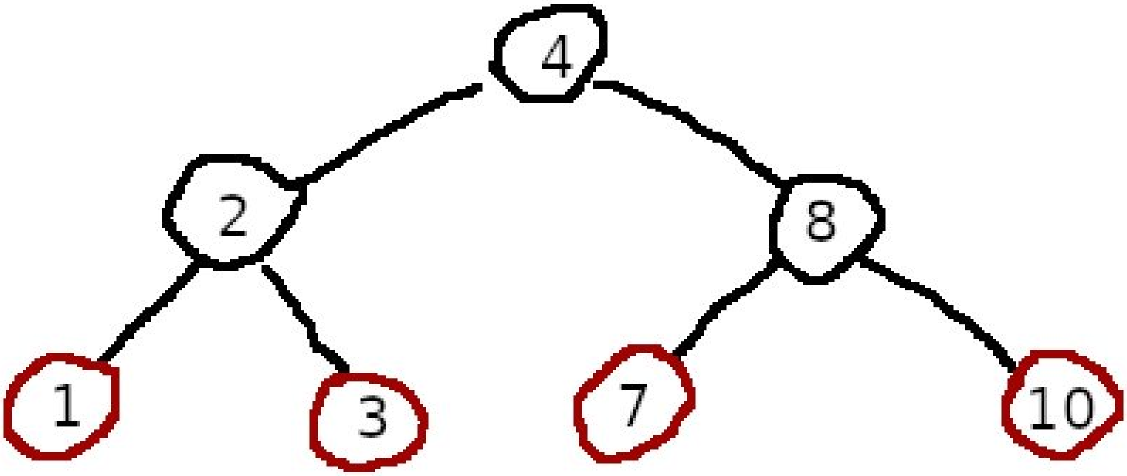
First rotation:



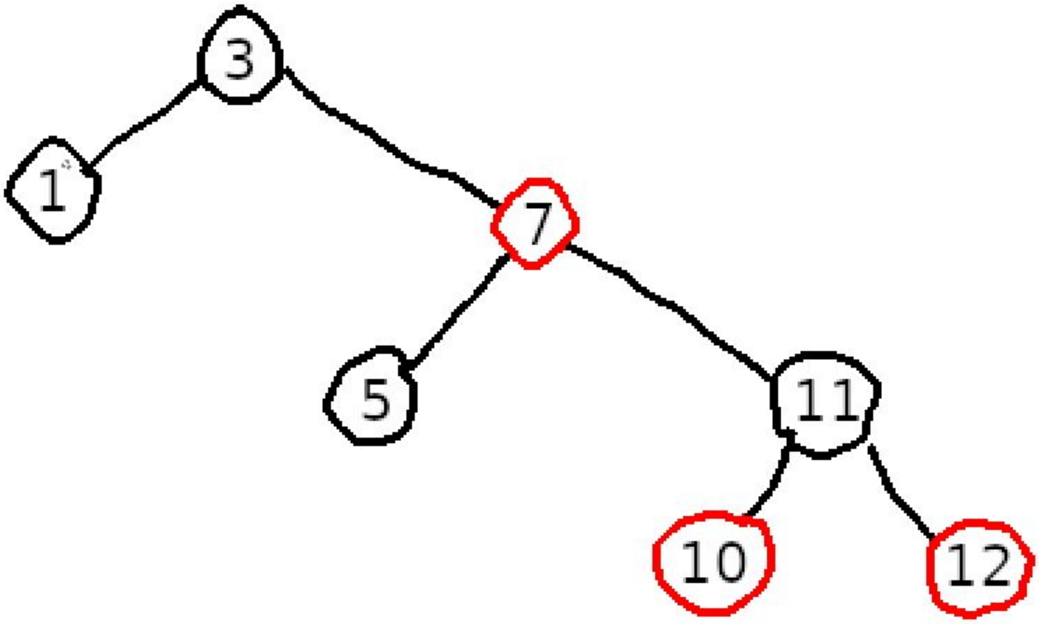
Second rotation:



8)

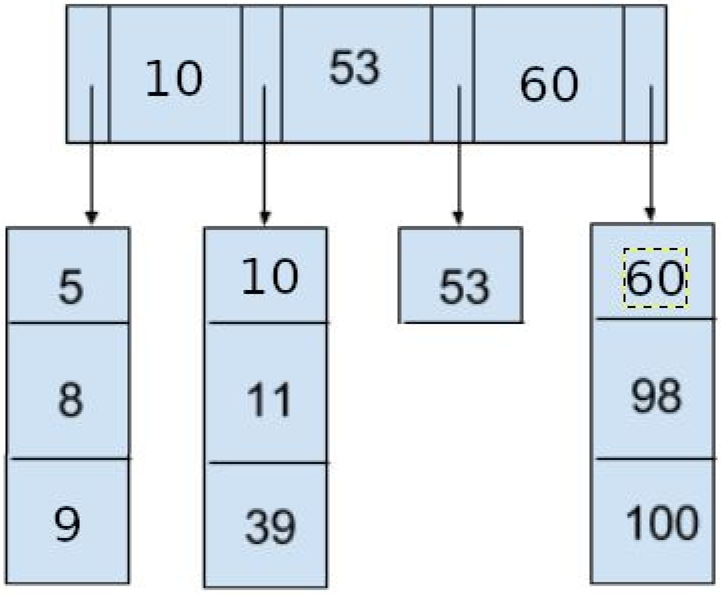


9)

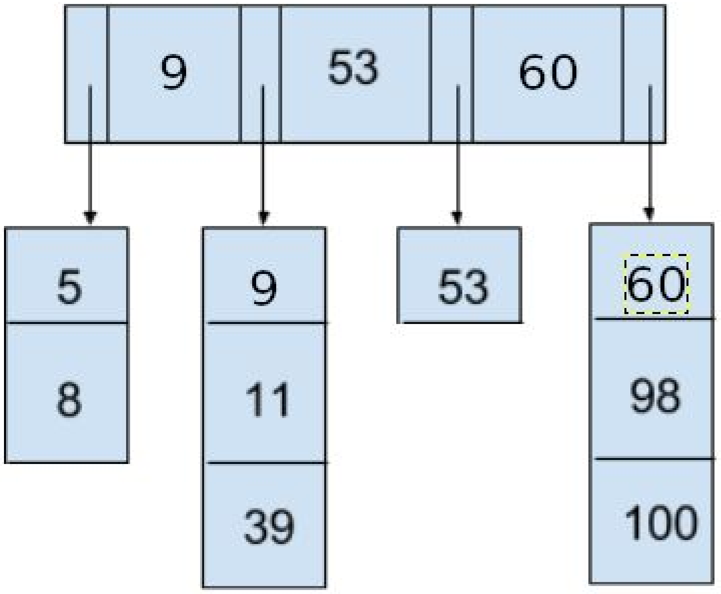


10)

a)



b)



11)

a)

Size of internal node = 5 * 128 bit UUID + 8 bit char + 16 bit int = 644 bits.

b)

Size of the leaf node is 5 * 152 bits + size of pointer. It's a 64 bit system, so the pointers are 64 bit. Thus the leaf nodes are of size 824.

c)

Height = $1 + \log_{\lceil m/2 \rceil}\left(\frac{N+1}{2}\right)$

N = number of nodes
m = number of children per node.

30,000 CustomerData records, 5 per node = 6,000 nodes.
Each node can have 5 children.

H = 1 + $\log_{2.5}$ (6,001 / 2) = 9.7379845085339 = 10

Height with 30,000 CustomerData records is 10.

d)

2,500,000 CustomerData records, 5 per node = 500,000 nodes.
Each node can have 5 children.

$$H = 1 + \log_{2.5} (500{,}001 / 2) = 13.564710156371 = 14$$

Height with 2,500,000 CustomerData records is 14.