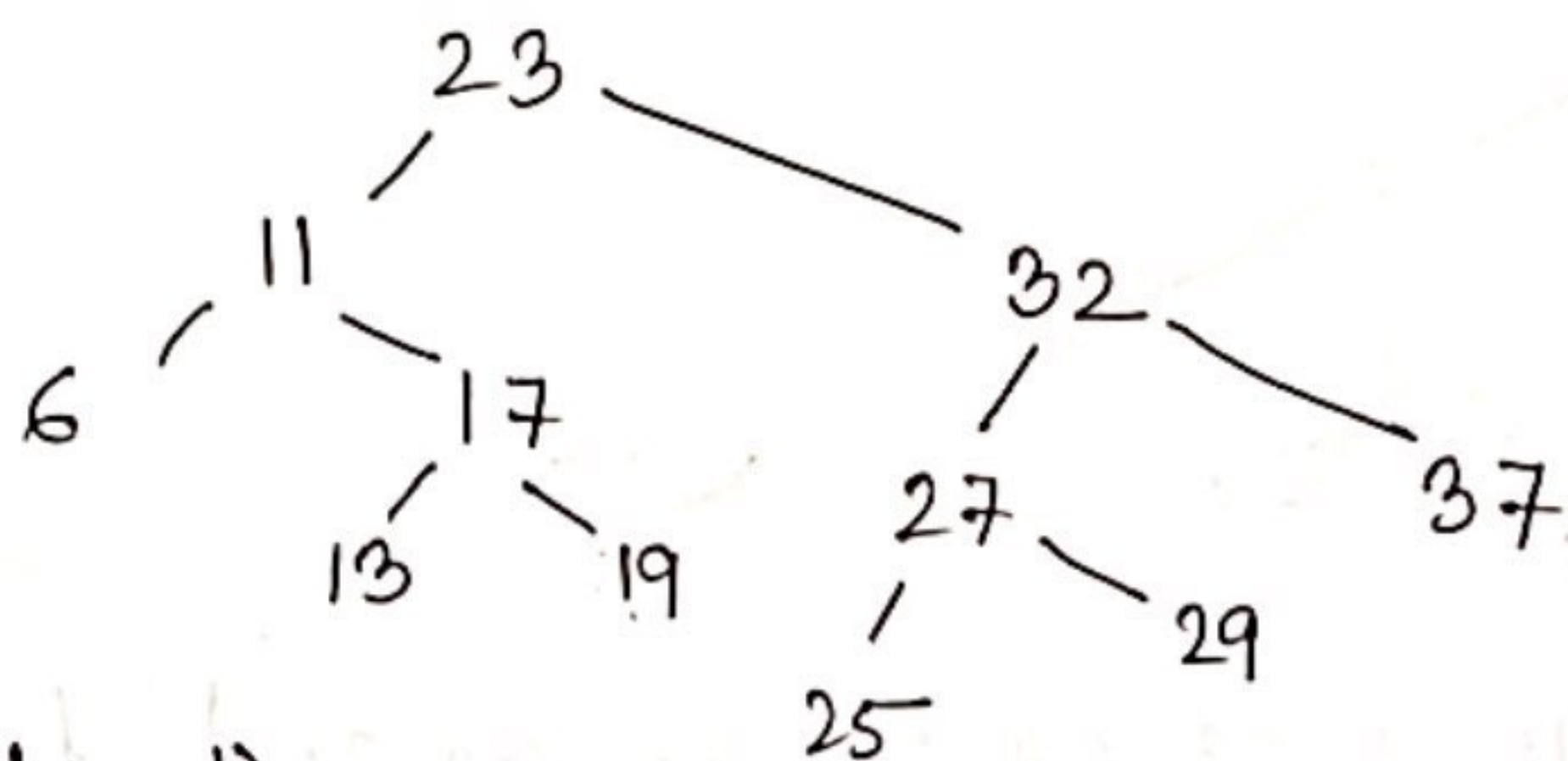


① [23, 11, 17, 32, 37, 27, 25, 19, 6, 13, 29]

a)

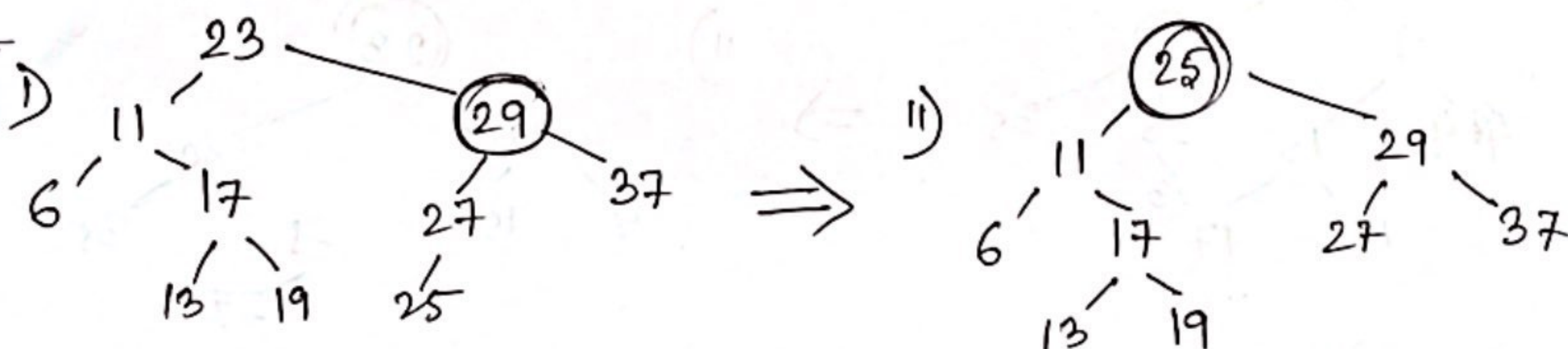


b) post (l → right → root):

6, 13, 19, 17, 11, 25, 29, 27, 37, 32, 23 | $\text{depth}(17) = 2$

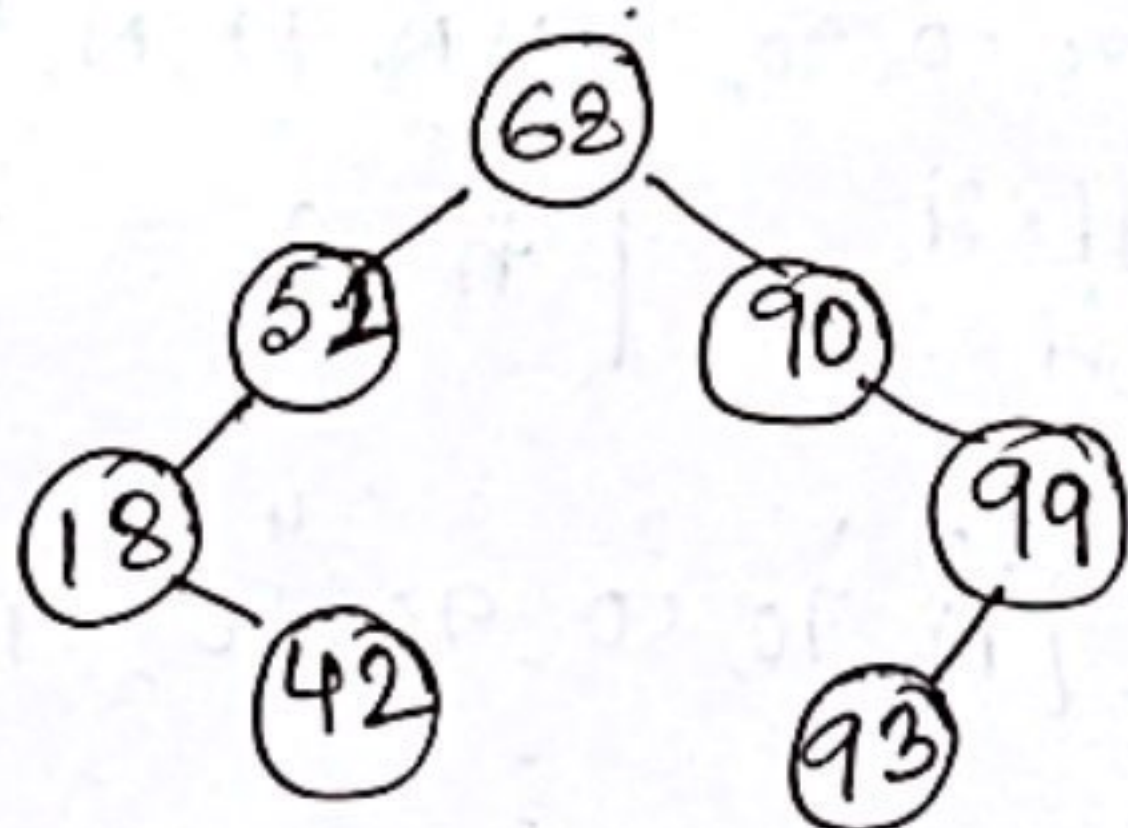
c) full, not complete, not perfect, balanced

d)



② [N, 68, 51, 90, 18, N, N, 99, N, 42, N, N, N, N, 93, N]

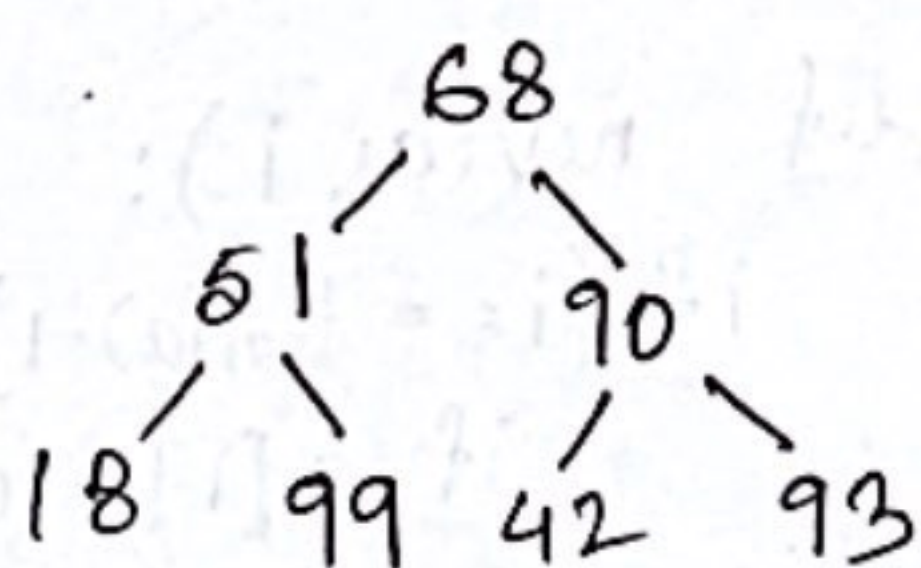
a)



left = 2i
right = 2i + 1

height of tree = 3

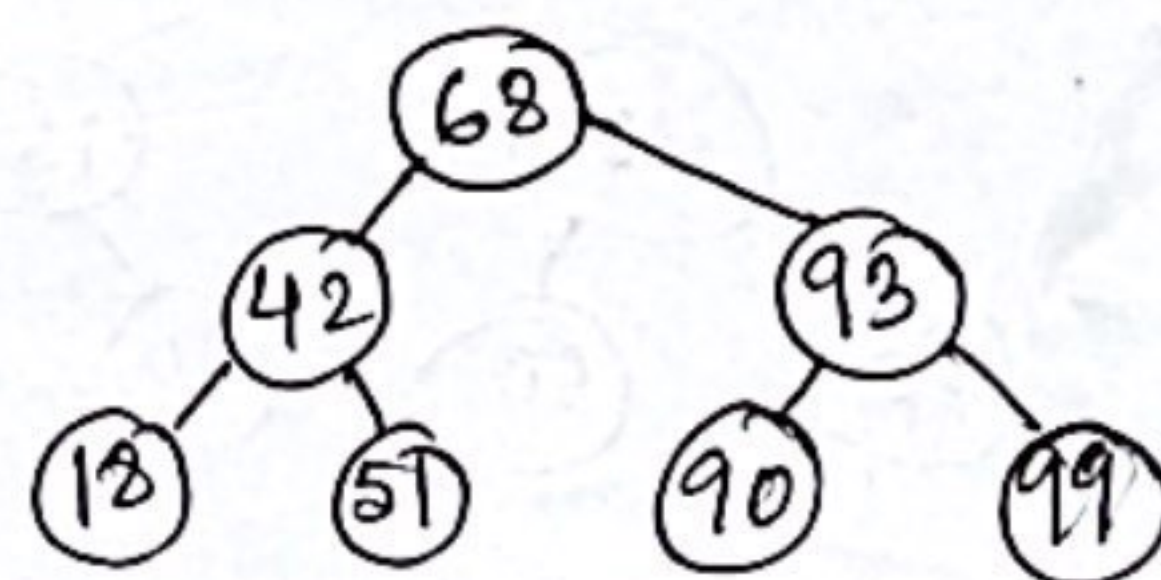
b) [N, 68, 51, 90, 18, 99, 42, 93]



c) Already balanced for (b) as each node has a height diff. ≤ 1 .

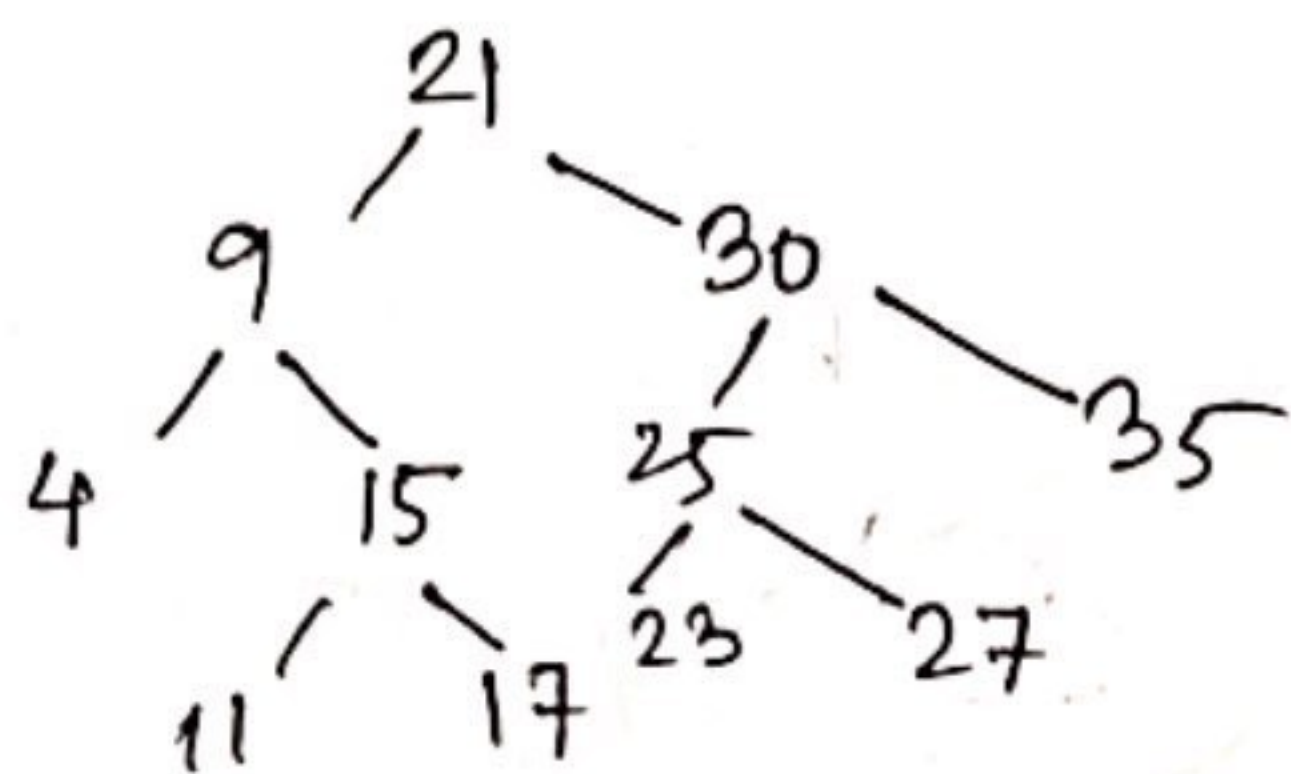
for a

in order: 18, 42, 51, 68, 90, 93, 99.
0 1 2 3 4 5 6



① [21, 9, 15, 30, 35, 25, 23, 17, 4, 11, 27]

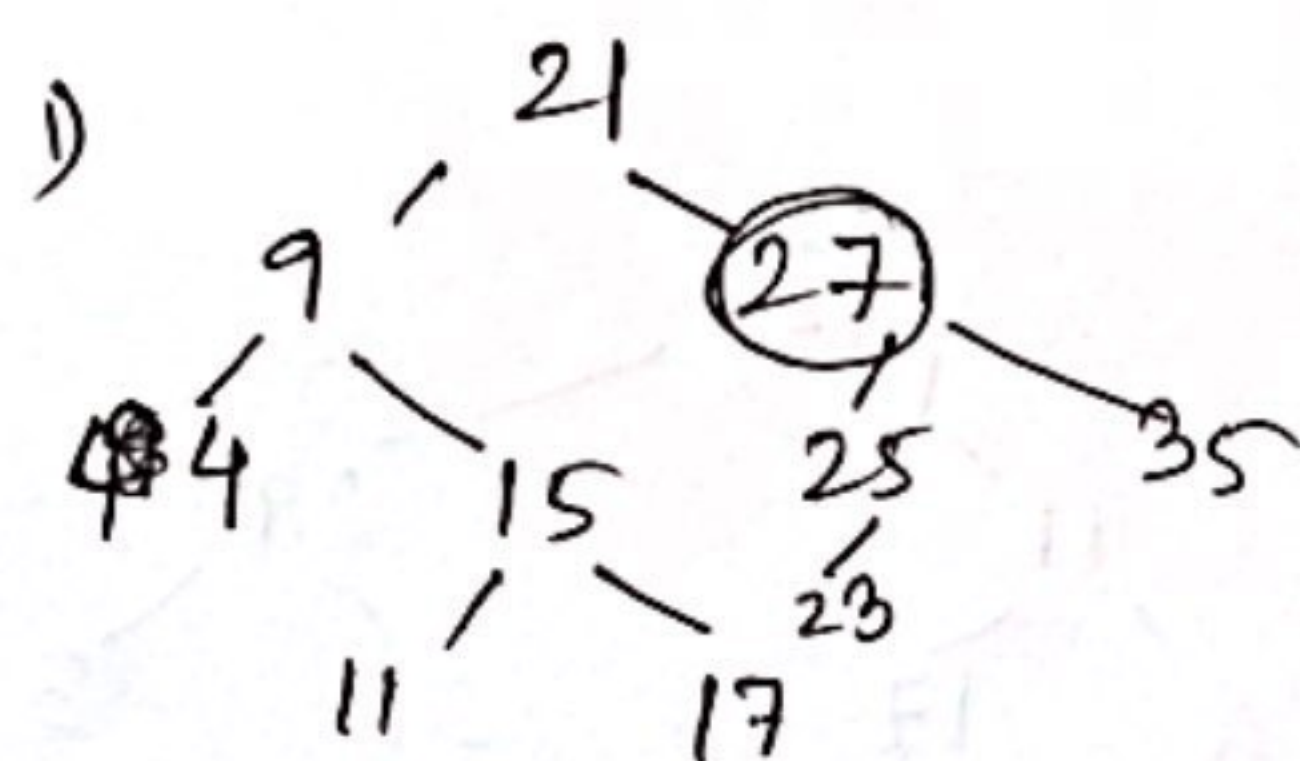
a)



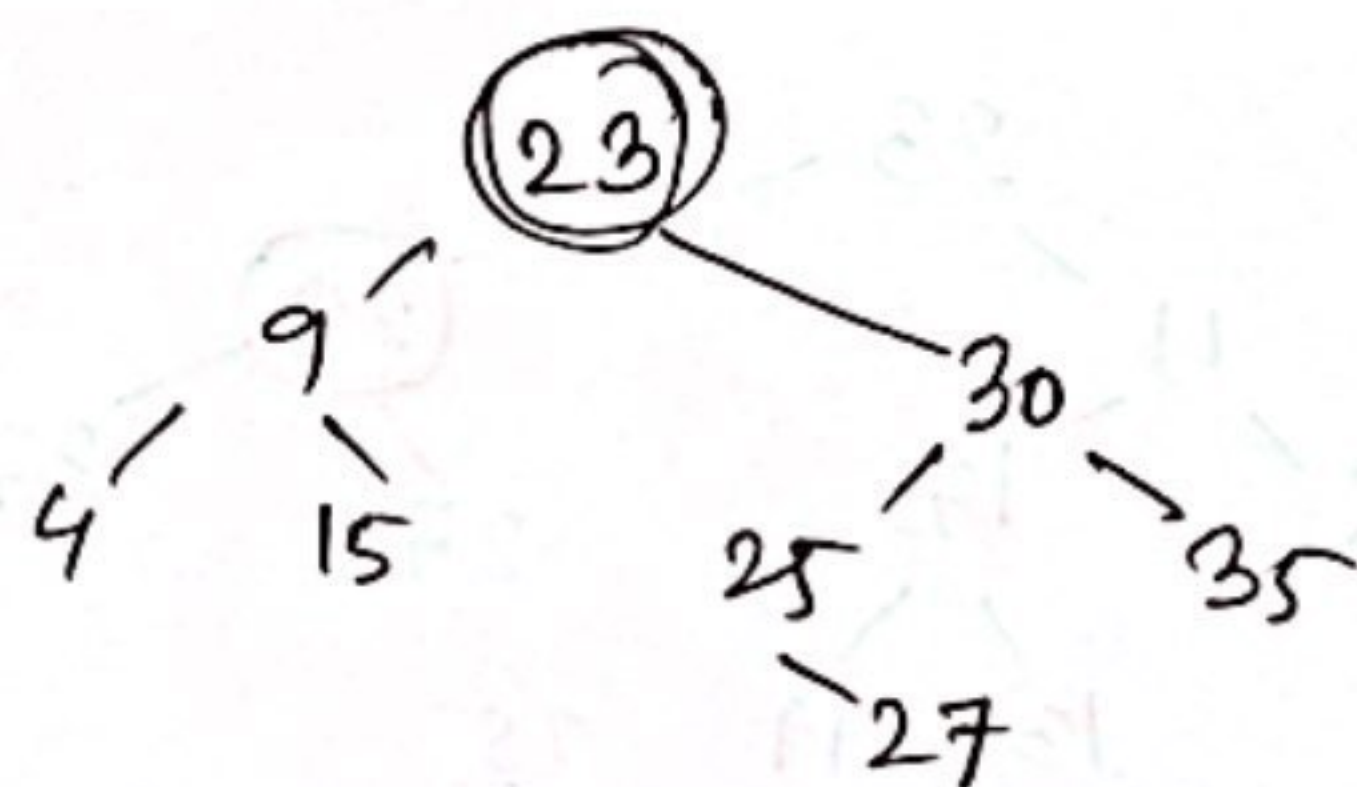
b) post : 4, 11, 17, 15, 9, 23, 27, 25, 35, 30, 21 | $d(15) = 2$

c) full, not complete, not perfect, balanced.

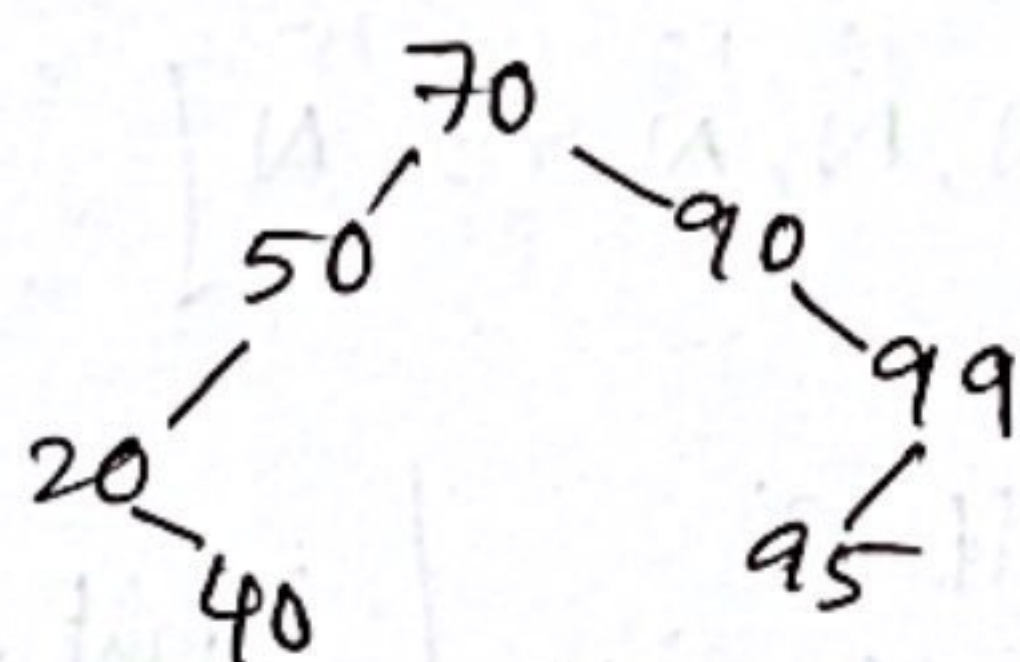
d)



\Rightarrow

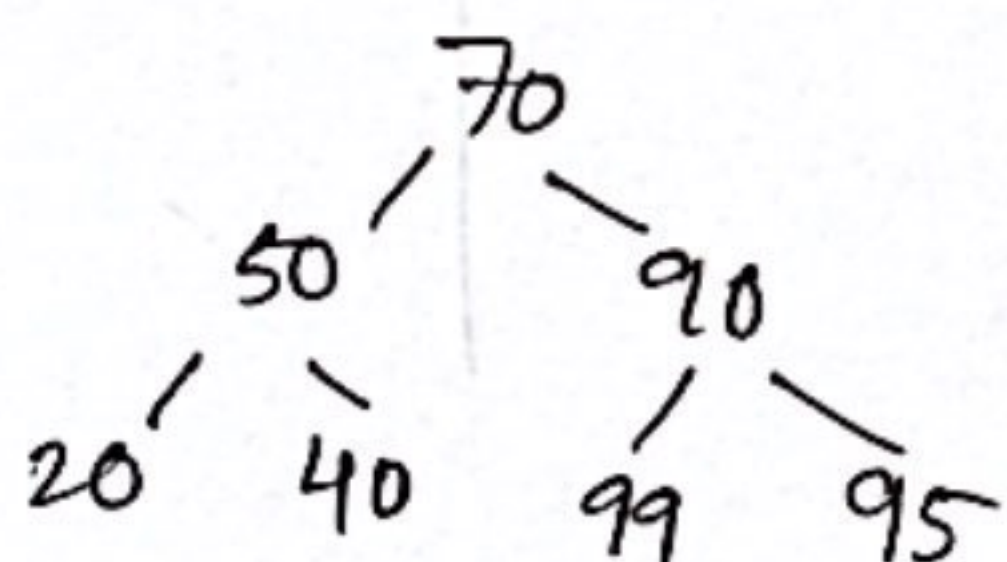


② a



$[N, 70, 50, 90, 20, N, N, 99, N, 40, N, N, N, N, 95, N]$
 $left = 2i$ | $h = 3$
 $right = 2i + 1$

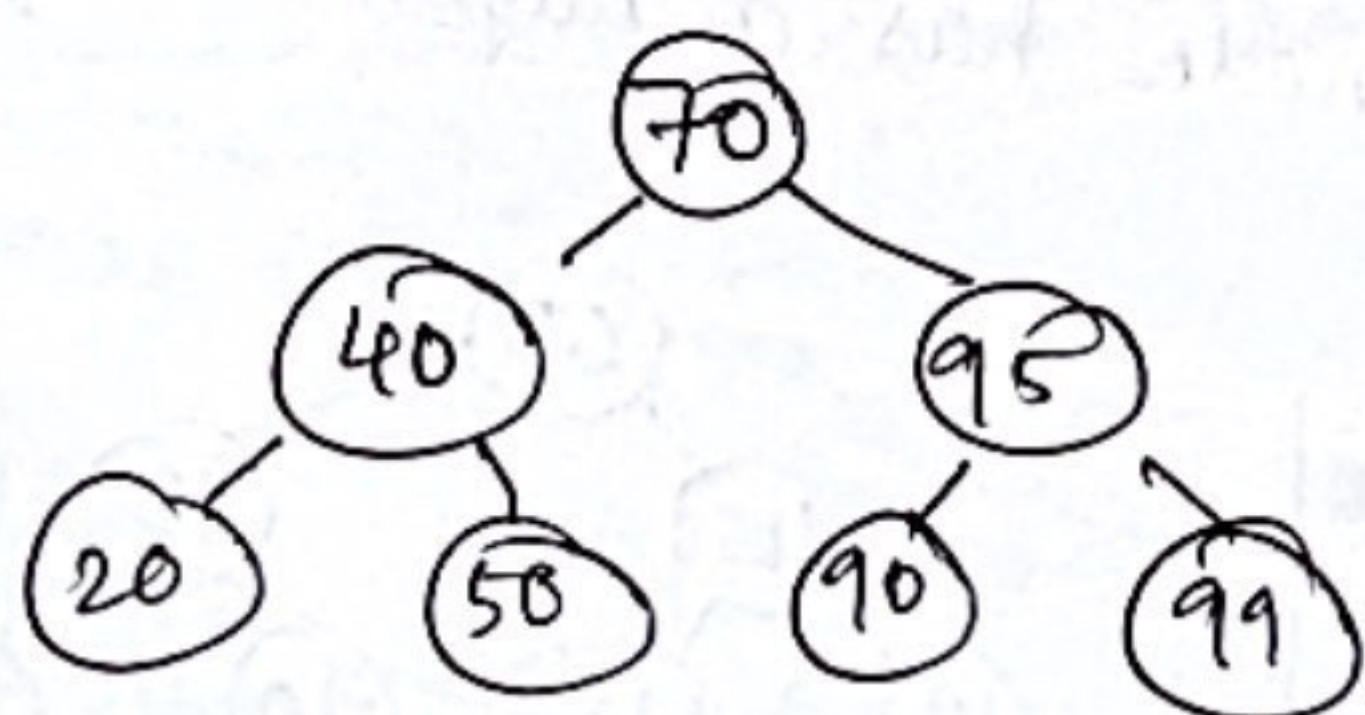
b.



$\leftarrow [N, 70, 50, 90, 20, 99, 40, 95]$

c. balanced. for (b)

for a. in order: $20, 40, 50, 70, 90, 95, 99$



Recursion

```
def rev(a, i):
    if (i == len(a) - 1):
        if a[i] >= 0 and a[i] % 2 != 0:
            print(a[i])
    else:
        rev(a, i + 1)
        if a[i] >= 0 and a[i] % 2 != 0:
            print(a[i])
```