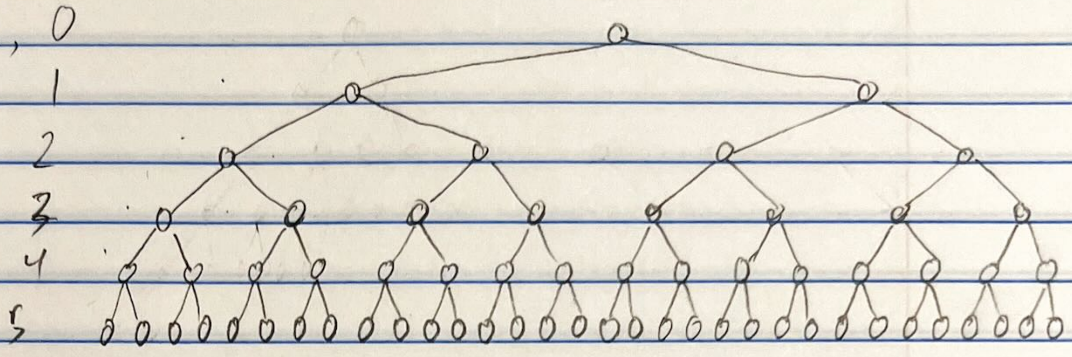
R-2.7 binary tree with n node use vector S, p be the level

```
Algorithm root()
    return S.elementAtRank(1)

Algorithm parent(p)
    return p./2

Algorithm leftChild(p)
    return 2p

Algorithm rightChild(p)
    return 2p+1

Algorithm isInternal(p)
    return (2*p+1 <= S.size)

Algorithm isExternal(p)
    return (2*p + 1 > S.size)

Algorithm isRoot(p)
    return p == 1;
```

R-2.8 a,



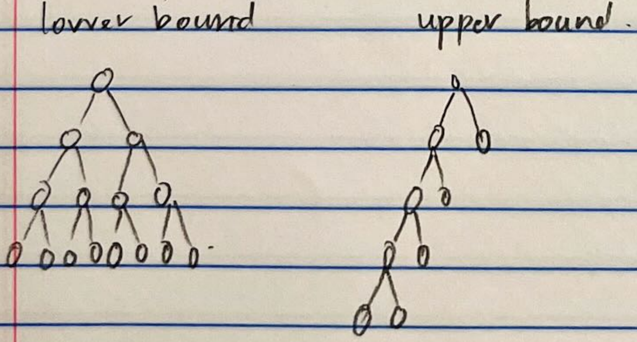
b, Minimum number of external node for a binary tree with height h? h+1

c, Maximum number of external node for a binary tree with height h? 2^h

d, $\log(n+1)-1 \leq h \leq (n-1)/2$

lowerbound when T is perfect binary tree.

upperbound when only child node has child



```
Algorithm Shuffle (S)
    n := S.size()
    while n > 0 do
        i := randomInt(n)
        swap (S, i, n-1)
        n := n-1

Algorithm swap (S, a, b)
    temp := S.elementAtRank(a)
    S.replaceElement(a, b)
    S.replaceElement(b, temp)
```