Name: Md Habibur Rony
Student ID: 984582
Weekday: Week 1- Day 3

<u>Answer to the Q. No. R-2.7:</u>

Algorithm Root()
Output: index of root
return S[1]

Algorithm Parent(i)
return S[i/2]

Algorithm LeftChild(i)
return S[i*2]

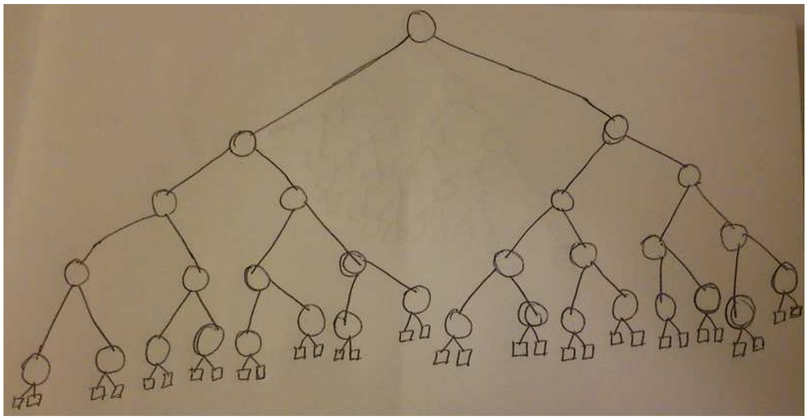
Algorithm RightChild(i)
return S[(i*2)+1]

Algorithm IsIternal(i)
If LeftChild(i)! = null or RightChild(i) != null Then
    return True
Else
    return False

Algorithm IsExternal(i)
If LeftChild(i) = null And RightChild(i) = null Then
    return True
Else
    return False

Algorithm IsRoot(i)
If i = 1 Then
    return true
Else
    return false

<u>Answer to the Q. No. R-2.8(a):</u>

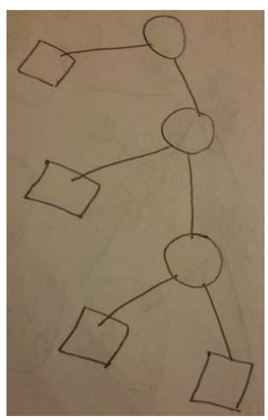


Answer to the Q. No. R-2.8(b):

We know that the range of external node is $h+1 < e < 2^h$

We can say that a binary tree is unbalance only when it has only left side child nodes or right side child nodes. Because either left side or right sides of the root node contains one external node. In this case, the minimum number of external nodes will be: $e=h+1$.

For example,



If $h = 3$ then minimum number of external nodes will be: $e=3+1 = 4$.

Answer to the Q. No. R-2.8(C):

We can say that a binary tree is balance only when it has left side child nodes and right side child nodes. Because both of the left side and right sides of the root node contains maximum numbers external nodes. In this case, the maximum external nodes will be: $e=2^h$.

For example,

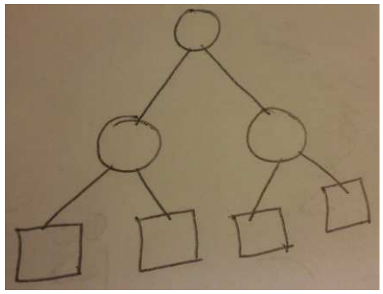


Fig 2.8.c

If h = 2 then maximum number of external nodes will be: $e=2^2 = 4$.

Answer to the Q. No. R-2.8(d):

From above the fig of 2.8.c we know i=3 and e = 4.

$$\log(n+1) - 1 \le h \le \frac{n-1}{2}$$
$$\Rightarrow \log(i+e+1) - 1 \le h \le \frac{i+e-1}{2}$$
$$\Rightarrow \log(3+4+1) - 1 \le h \le \frac{3+4-1}{2}$$
$$\Rightarrow \log 8 - 1 \le h \le \frac{6}{2}$$
$$\Rightarrow 2 - 1 \le h \le 3$$
$$\Rightarrow 1 \le h \le 3$$

So we can say that the height of binary tree which is balance or unbalance will be in between $\log(n+1) - 1 \le h \le (n-1)/2$

Answer to the Q. No. R-2.8(e):

Say, n=1 and h=1 then lower and upper bounds on h be attained with equality.

So, $\log(n+1) - 1 \le h \le (n-1)/2$

$\Rightarrow \log(1+1)-1 \le h \le (1-1)/2$

$\Rightarrow 1-1 \le h \le 0/2$

$\Rightarrow 0 \le h \le 0$

Answer to the Q. No. R-2.2:

We know that the running time for push and pop operations for stacks are O(1). Amortized running time of each operation is O(1) this mean O(n) is the total running time. Amortized cost of 1 operation is O(n)/n=O(1)

Answer to the Q. No. R-2.7:

The running time is nlogn

Algorithm RandomCards(deckSeq)
i <- 0
While i < deckSeq Do
        r <- randomInt(deckSeq)
        deckSeq.SwapElement(deckSeq.atRank(i), deckSeq.atRank(r))
        i <- i + 1
End while
Return S
	
	O(1)
	O(n)
	O(n)
	For linked list $O(n^2)$
	O(n)
	
	O(1)
	$T(n)= O(n^2)$