BITS ID NO. - 2019ht13096

COURSE TITLE - SSZG519

NAME - BRAJ KISHOR

COURSE NAME - DATA STRUCTURES AND ALGORITHM DESIGNS

Q.1

(a)

Algorithm: Search (A, n, e)

Input: An array A of n integers
An element e

Output: Index of element e, if present
-1 , if not present.

```
currentIndex ← 0
while currentIndex < n
    if A[currentIndex] = e then
        return currentIndex

return -1
```

BITS ID NO - 2019ht13096 COURSE TITLE - SSZG519

NAME - BRAJ KISHOR COURSE NAME - DATA STRUCTURES AND ALGORITHM DESIGNS

Q.1

(b)

	Operation	freq
Current Index ← 0 —	1	1
while Current Index < n —	1	n+1
if A[currentIndex] = e —	2	n
return current Index —	1	1
Current Index ← current Index + 1 —	2	n
return -1 —	1	1

∴ Total no of primitive ops = 8

(c)

$T(n) = (1 \times 1) + 1(n+1) + 2n + (1 \times 1) + 2n$

As return statement will execute only once

$\therefore T(n) = 1 + n + 1 + 2n + 1 + 2n$

$\boxed{\therefore T(n) = 5n + 3}$

$f(n) = 5n+3$ is big-oh, if $f(n) = O(g(n))$

$f(n) = 5n+3, \quad g(n) = n$

$f(n) \le c\,g(n)$ for $n \geq n_0$

$5n+3 \le cn$

if $c = 6$, then $5n+3 \le 6n$

$\therefore n \geq 3.$

so; $5n+3 \le 6n$, if $n \geq 3$.

$\boxed{\therefore T(n) = O(n)}$, if $n \geq 3$.

Q.2

Given,

$$T(n) = 2T(n-1) \quad ; \text{ if } n > 1$$

$$T(1) = 1 \quad ; n = 1$$

$$T(n) = 2T(n-1) \quad \text{— } 1$$

for $(n-1)$, this eq. becomes

$$T(n-1) = 2T([n-1]-1)$$

$$T(n-1) = 2T(n-2)$$

Substitute this value in eq 1

$$T(n) = 2 \times (2T(n-2))$$

or, $T(n) = 4T(n-2)$ — 2

similarly for $T(n-2)$

$$T(n-2) = 2T([n-2]-1)$$

$$T(n-2) = 2T(n-3)$$

Q.2

Substituting the above value in eq (2)

$$T(n) = 4T(2T(n-3))$$

$$T(n) = 8T(n-3) \quad - (3)$$

So, in general

$$T(n) = 2^i T(n-i) \quad - (4)$$

Taking the base case, here

$$T(1) = 1$$

$$\therefore n-i = 1$$

$$\Rightarrow i = n-1 \ \& \ n = i+1$$

Substituting in eq (4)

$$T(n) = 2^{n-1} T(i+1-i)$$

$$\therefore T(n) = 2^{n-1} T(1)$$

$$\therefore T(n) = 2^{n-1} \times 1$$

$$\therefore T(n) = O(2^{n-1}) \simeq O(2^n)$$

$$\boxed{\therefore T(n) = O(2^n)}$$

BITS ID NO.- 2019ht13096
NAME- BRAJ KISHOR

Q.3

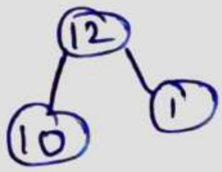
Operation	Q[0]	Q[1]	Q[2]	Front(f)	Rear(r)	Error
ENQUEUE(A)	A			0	1	No err
DEQUEUE	NULL			1	1	No err
DEQUEUE	NULL			1	1	EMPTY
ENQUEUE(B)	NULL	B		1	2	No err
ENQUEUE(X)	NULL	B	X	1	3	No err
ENQUEUE(Z)	NULL	B	X	1	3	FULL
DEQUEUE	NULL	NULL	X	2	3	No err

Q. 4

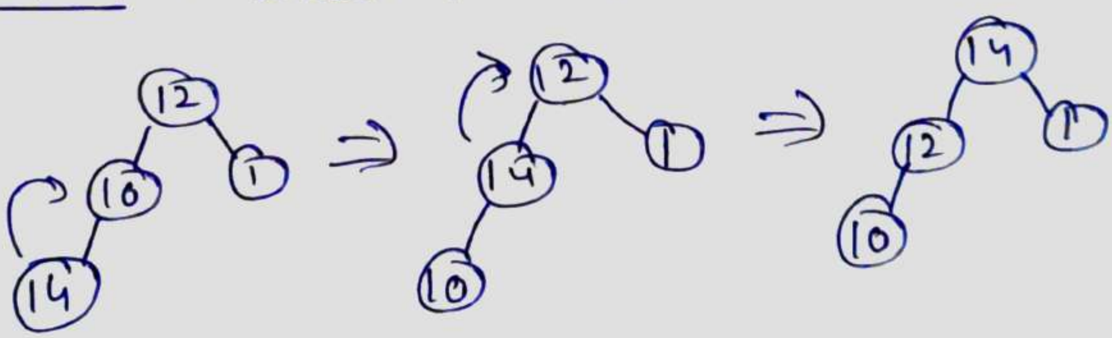
STEP. 1 : Insert 10

STEP 2 : Insert 12

Step 3 : Insert 1

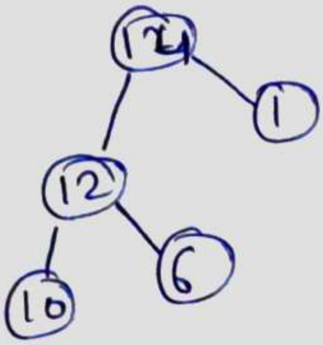


Step 4 : Insert 14

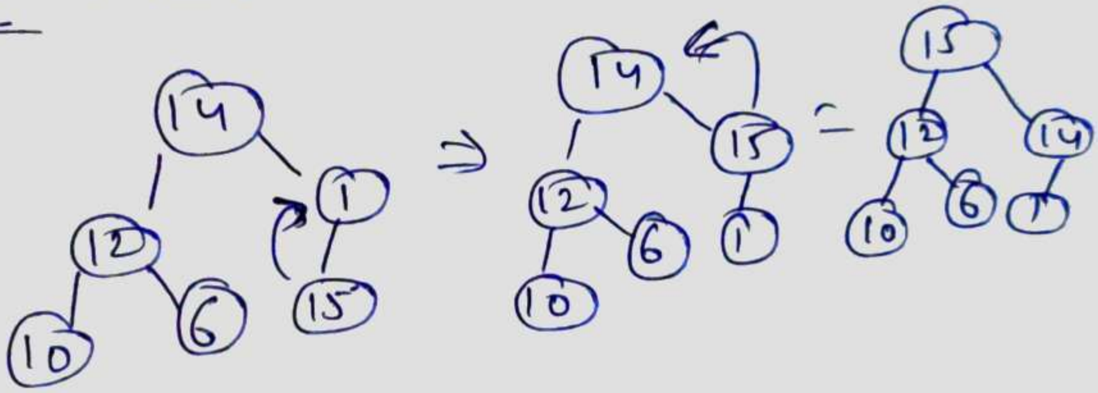


Q.4

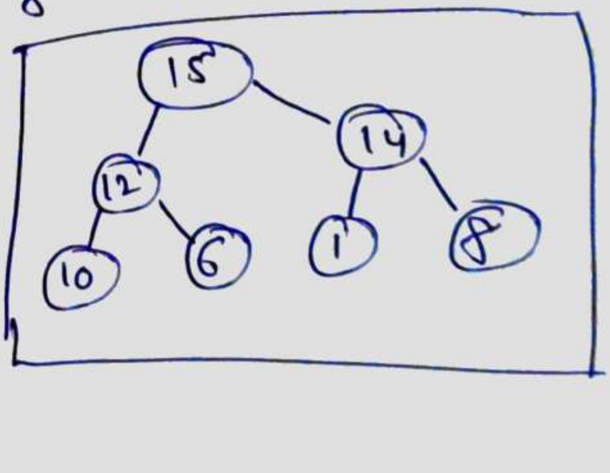
STEP 5: Insert 6



STEP 6 : Insert 15



STEP 7 : Insert 8



Q.5

(a) Given a tree of height 'h'

(i) Internal nodes

minimum $= h$
maximum $= 2^h - 1$

(ii) External nodes

minimum $= 1$
maximum $= 2^h$

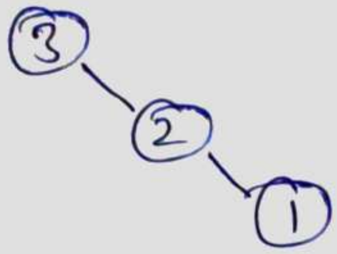
(iii) Total nodes

minimum $= h+1$
maximum $= 2^{h+1} - 1$

Q.5

(b)

Min Case



Internal nodes

3, 2

total = 2.

$\boxed{\therefore h = 2}$

External nodes

1.

total = 1

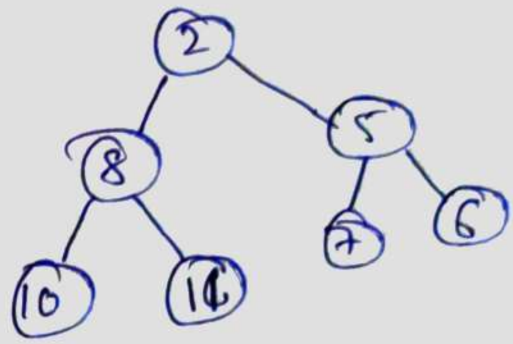
$\boxed{\therefore h = 1}$

Total nodes

3, 2, 1 $\therefore$ Total = 3

$\boxed{\therefore \text{Nodes} = h + 1 = 2 + 1 = 3}$

Max Case



Internal nodes

2, 8, 5

total = 3.

$$\boxed{\begin{aligned}&\therefore 2^h - 1\\ &= 2^2 - 1 = 3.\end{aligned}}$$

External nodes

10, 11, 7, 6

total = 4.

$\boxed{\text{i.e. } 2^h = 2^2 = 4}$

Total Nodes

2, 8, 5, 10, 11, 7, 6

Total = 7

$\boxed{\text{Nodes} = 2^{h+1} - 1 = 2^3 - 1 = 7}$

THE END