

Trees

Question 7

Number of nodes = N

Number of nodes at level k = 2^k

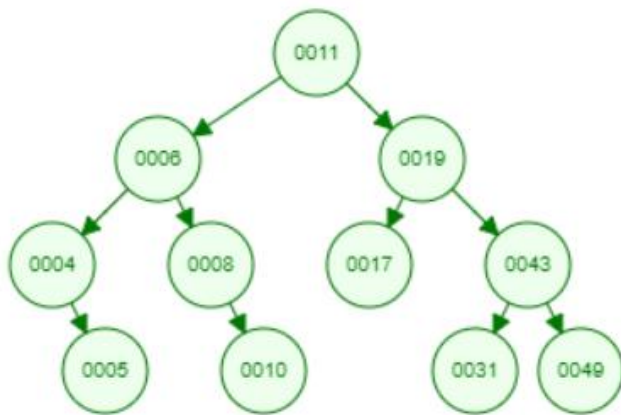
Number of levels = h

$$\sum_{i=0}^h 2^i = N$$

$$2^{h+1} = N$$

$$h = \log_2(N) - 1 = O(\log(N))$$

Question 8



Question 10

```

FUNCTION searchNode(Tree,value):
    while Tree != NULL and Tree.value != value:
        if value > Tree.Value:
            Tree <- Tree.right
        else:
            Tree <- Tree.left
    if root == NULL:
        return False
    return True
  
```

Question 11

```

FUNCTION searchWithParentsReturn(root,value):
    rootParent <- root
    while root != NULL and root.value != value:
        rootParent <- root
        if value < root.value:
            root <- root.right
        else:
            root <- root.left
    if root == NULL:
        return False ,None, rootParent
    return True , root,rootParent
  
```

```

FUNCTION getNextBigger(root):
    root <- root.right
    rootOld <- root
    while root != NULL:
        rootOld <- root
        root <- root.left
    return rootOld
  
```


Heaps

Question 1:

A maximum heap is a binary tree where the value of the parent is bigger than both of his children, thus having the maximum value of the tree at top.

A minimum heap is a binary tree where the value of the parent is smaller than both of his children, thus having the minimum value of the tree at top.

It is possible to have minheap and BST is we allow for duplicates.

Question 2:

$$T(n) = \sum_{i=0}^{h-1} 2 \cdot 2^i (h-i)$$

$$= 2h \sum_{i=0}^{h-1} 2^i - 2 \sum_{i=0}^{h-1} 2^i i$$

$$2h \sum_{i=0}^{h-1} 2^i = 2h 2^h - 2h$$

$$2X = 2 \sum_{i=0}^{h-1} 2^i i = 2 (1 \times 2^0 + 2 \times 2^1 + 3 \times 2^2 + \dots + (h-1) 2^{h-1})$$

$$\frac{X}{2} = 2 (1 + 2 \times 2^1 + 3 \times 2^2 + \dots + (h-2) 2^{h-1})$$

$$= 2 (1 \times 2^1 + 2 \times 2^2 + \dots + (h-2) 2^{h-2} + 1 + 2 + 2^2 + 2^3 + \dots + 2^{h-1})$$

$$\frac{X}{2} = 2 \left(\frac{X}{2} - 2^{h-1} (h-1) + 2^{h-1} - 1 \right)$$

$$\therefore X = h 2^{h+1} - 2^{h+1} - 2^h + 2$$

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

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

Question 3:

