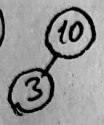
Tarık Berkan Bilge 22003392 Sec:1 Assignment: 3

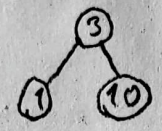
Heaps and AVL Tree

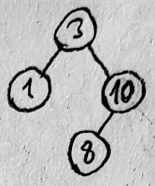
Question 1

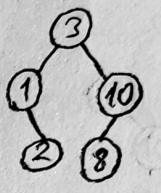
a)

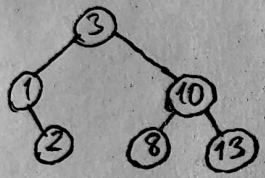
1) 

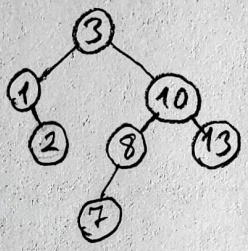
2) 

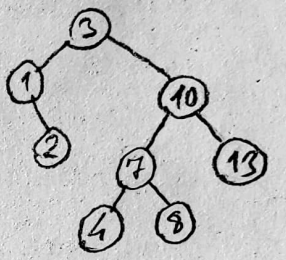
3) 

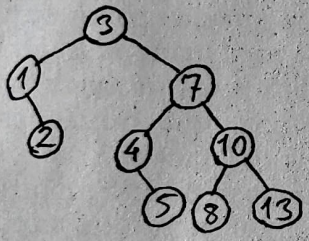
4) 

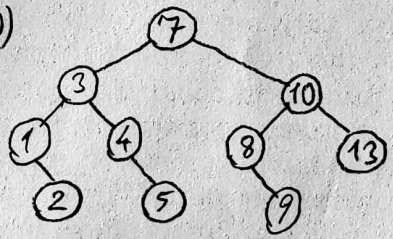
5) 

6) 

7) 

8) 

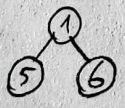
9) 

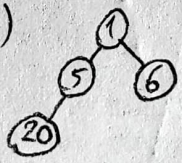
10) 

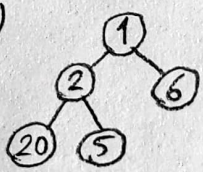
b)

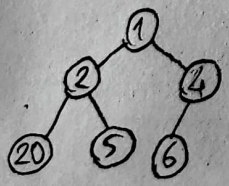
1) 

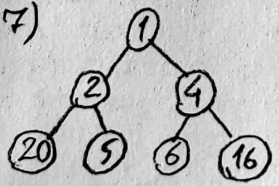
2) 

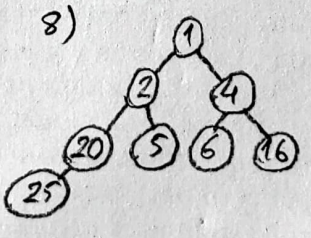
3) 

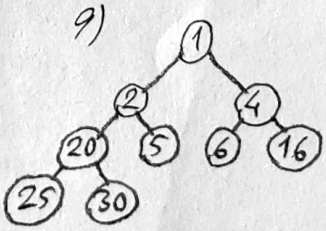
4) 

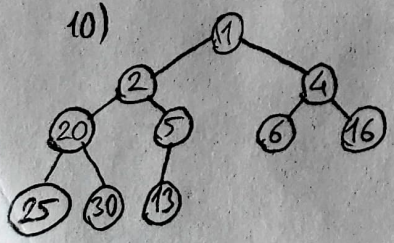
5) 

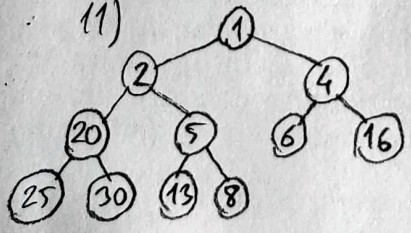
6) 

7) 

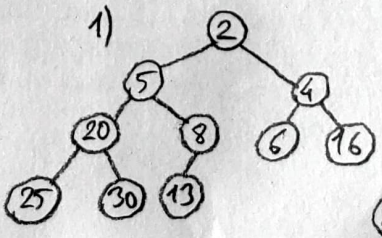
8) 

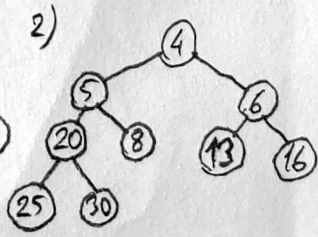
9) 

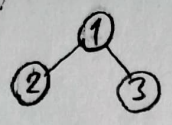
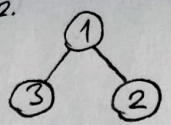
10) 

11) 

• Deletion

1) 

2) 

c) Lets consider these heaps: 1.  & 2. 

Preorder Traversal: For the 1. heap, result is "1,2,3" and it is sorted but for the 2. heap result is "1,3,2" and it is not sorted. Therefore, preorder traversal doesn't sort a heap.

Inorder Traversal: For the 1. heap, result is "2,1,3" and it is not sorted. Therefore, inorder traversal doesn't sort a heap.

Postorder Traversal: For the 1.heap, result is "2,3,1" and it is not sorted. Therefore, postorder traversal doesn't sort a heap.

In summary, none of the preorder, inorder, postorder traversals of a binar heap produce a sorted order. Binary heaps are designed to efficiently support heap operations, but their elements are not arranged in a way that provides a sorted order through standard tree traversals.

d) • It can be calculated using the recursive formula: $n(h) = n(h-1) + n(h-2) + 1$

• Based on the above, $n(15) = n(14) + n(13) + 1 = 1596$

$n(2) = n(1) + n(0) + 1$
$n(1) = 1$
$n(0) = 0$

12
7
4
2
1
0

e)

```
function isMinHeap(node) {
    if(node == NULL)
        return true;
    if(node.left != NULL && node.left.key < node.key)
        return false;
    if(node.right != NULL && node.right.key < node.key)
        return false;
    return isMinHeap(node.left) && isMinHeap(nod.right);
}
```