Cs 2005D Data structures and algorithms
Tutorial I - Winter 2019

1. Starting with an empty tree, construct an AVL tree by inserting the following keys in the order given: 2, 3, 5, 6, 9, 8, 7, 4, 1.

2. Starting with an empty tree, construct a Red-Black tree inserting the following keys in the given order: 23,21,16,14, 9, 10, 12, 17, 20.

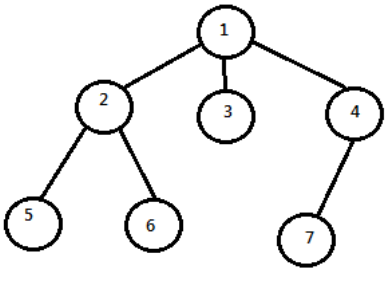
3. Consider the following code for finding the connected components of a graph G.

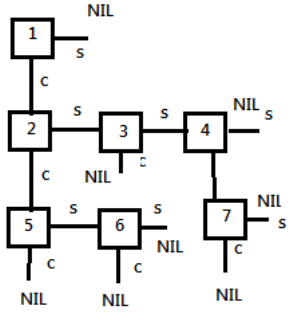
```
CONNECTED-COMPONENTS(G)
1. for each vertex v Є G.V
2.       MAKE-SET(v)
3. for each edge(u,v)Є G.E
4.     if FIND-SET(u) ≠ FIND-SET(v)
5              UNION(u,v)
```

if CONNECTED-COMPONENTS is called for a graph G=(V,E) with |V|=12, |E|= 15 and k=4 connected components , how many times would FIND-SET be called? How many times would UNION be called?

4. All the BSTs produced by arrangements of keys in ascending or descending order have only a single leaf. Is it possible to have a single leaf BST produced by any other key permutation? Justify your answer.

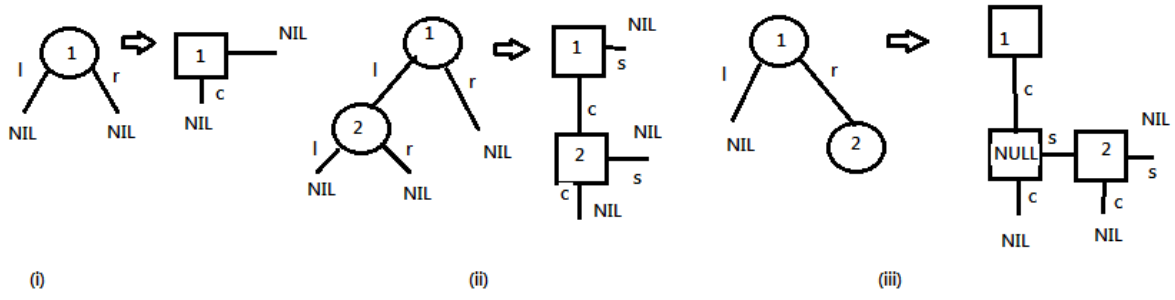
5. An alternative implementation for a tree is to have nodes with two pointers, child and sibling. The child pointer points to the first child, and the sibling pointer of every node points to the next sibling (child of its parent)
Example.



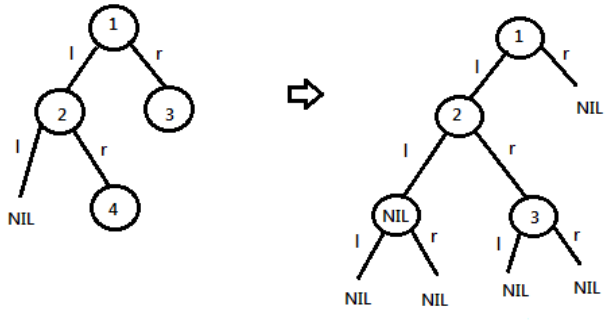


For a binary tree, you may assume a special "**null node**" for **an absent left child of a non leaf node**, which has null value as its content and a pointer to the right child of the parent.

Examples.



Write a recursive algorithm to convert a given binary tree to its new implementation, using the leftchild pointer as child and right child pointer as sibling pointers. [5]
Example:



6. Draw an AVL tree of height 3 that has the minimum number of nodes, that is, the minimum number of nodes among all AVL trees of height 3.

7. Draw the resultant Fibonacci heap if after doing the following operations on a initially empty Fibonacci heap H in the given order.
Insert(H,4), Insert(H, 45), Insert(H, 5), Insert(H, 15), Insert(H, 32), Insert(H, 2), Insert(H, 21), Insert(H, 53), Insert(H, 16), Insert(H, 9), Insert(H, 29), Insert(H, 17), Insert(H, 38), FIB-EXTRACT_MIN(H), DECREASE-KEY(H,45,1).