

CONCORDIA UNIVERSITY
COMPUTER SCIENCE AND SOFTWARE ENGINEERING

Data Structures and Algorithms
(COMP#5511)

ASSIGNMENT#2

PREPARED BY:

GROUP # 18

LIAN LONGFENG (40040689)
COLIN GALLACHER (40070588)
MOHAMED RASHED (40038347)

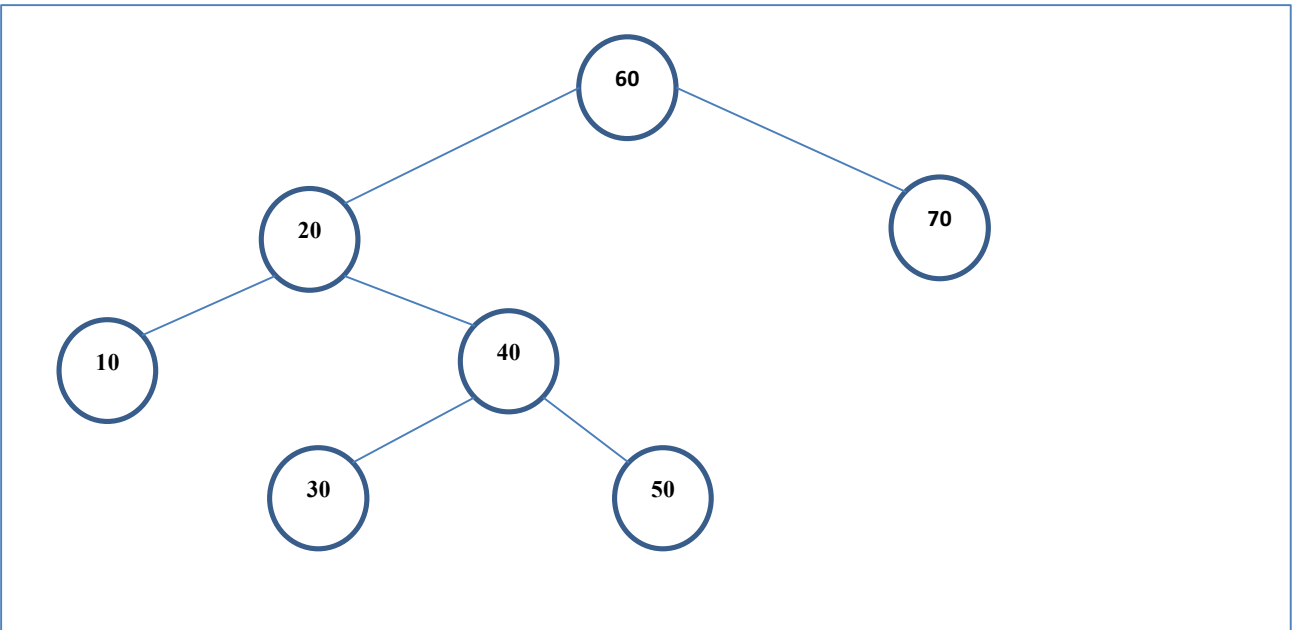
SUBMITTED TO:

PROFESSOR B. DESAI

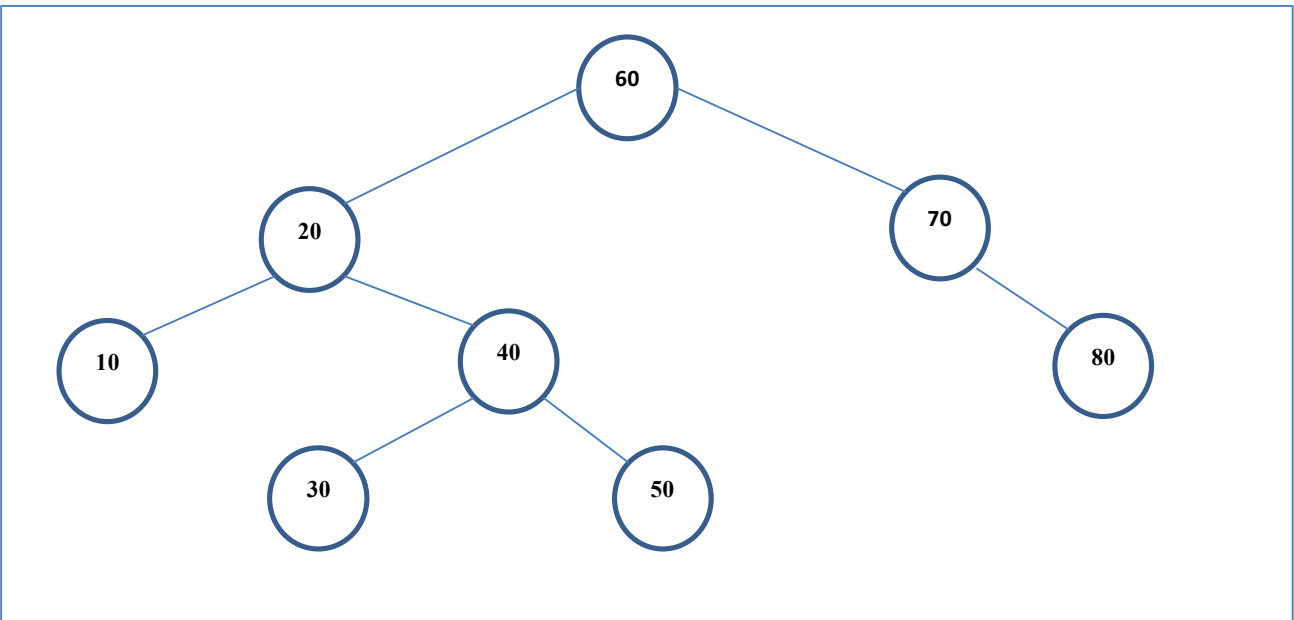
October 2017

1. a.

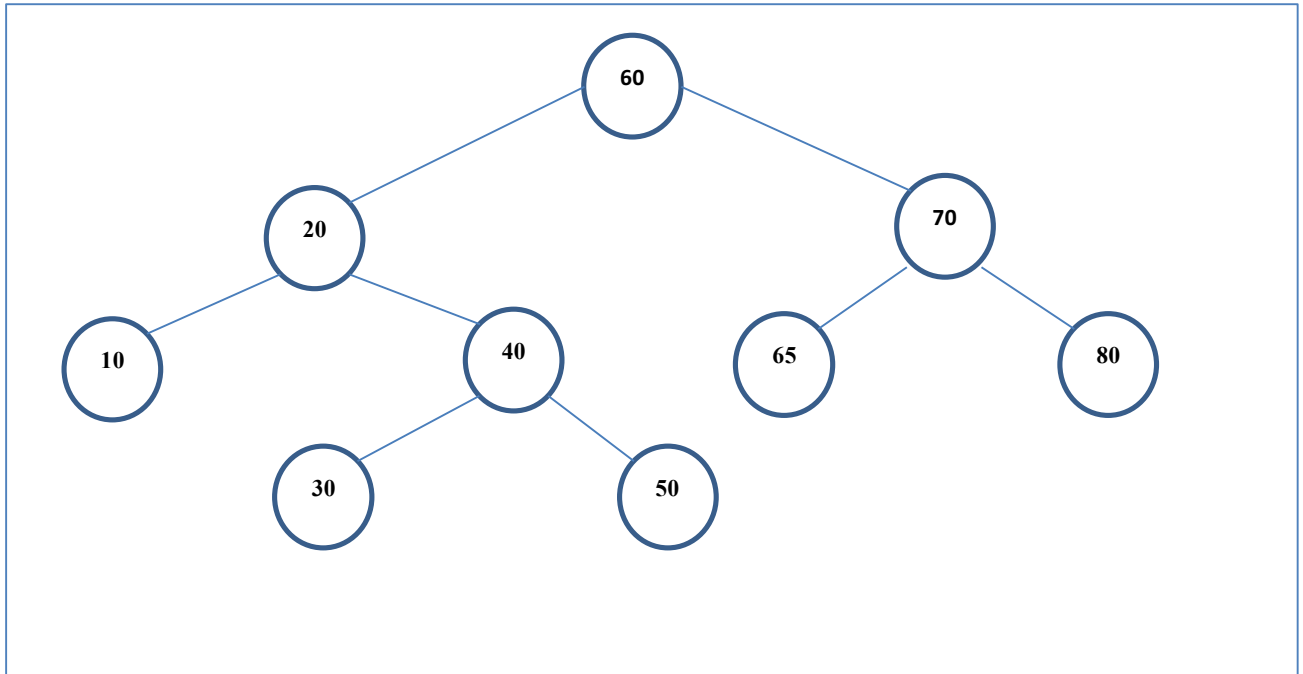
Initial Tree



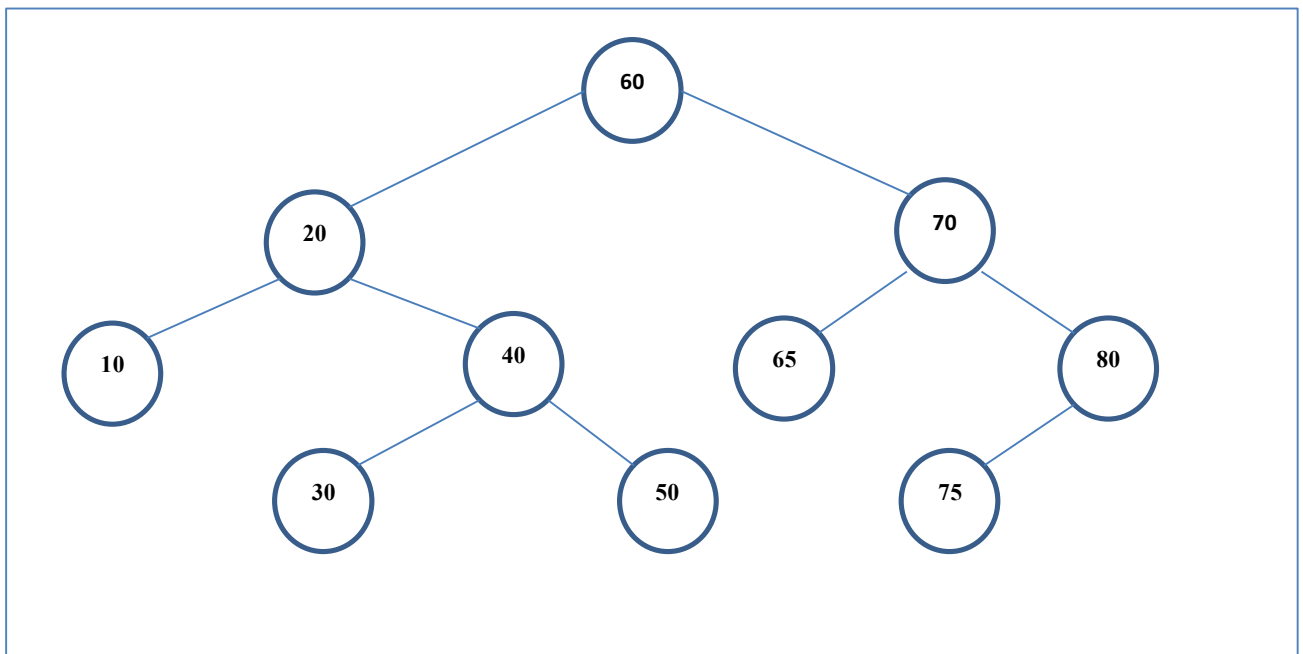
Step 1



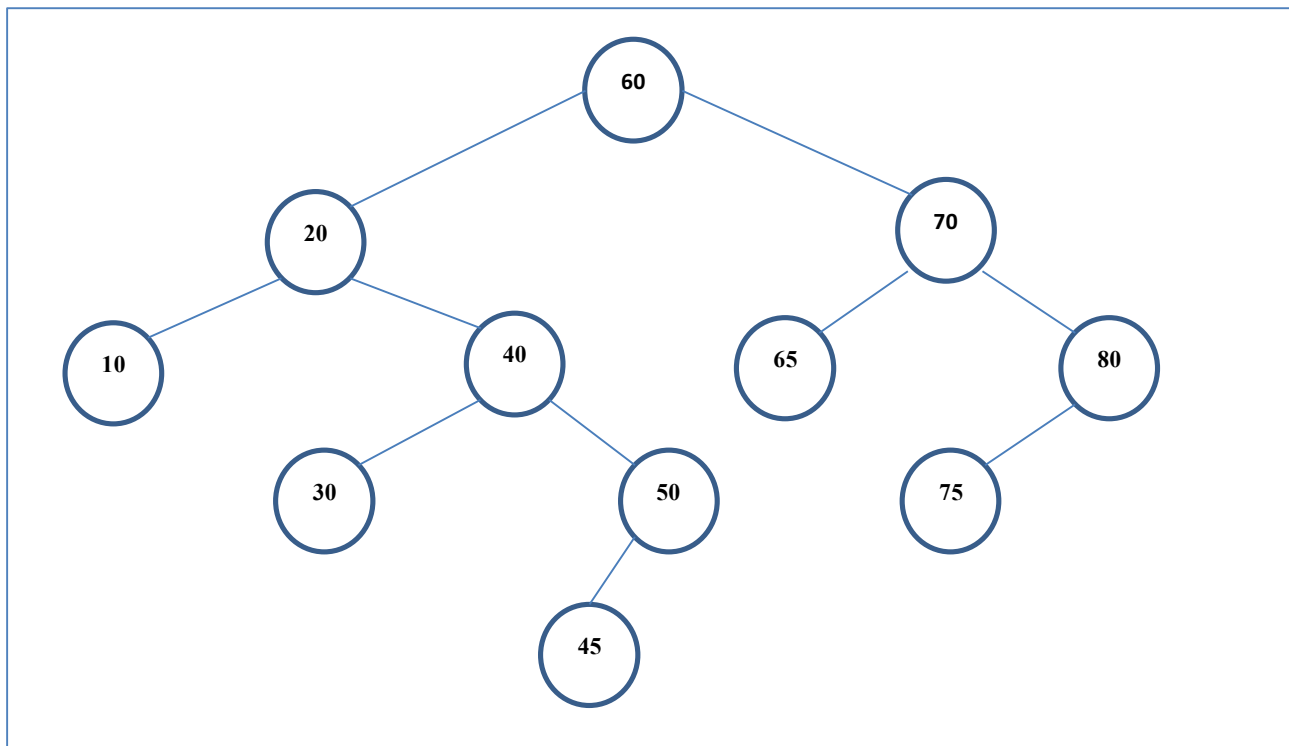
Step 2



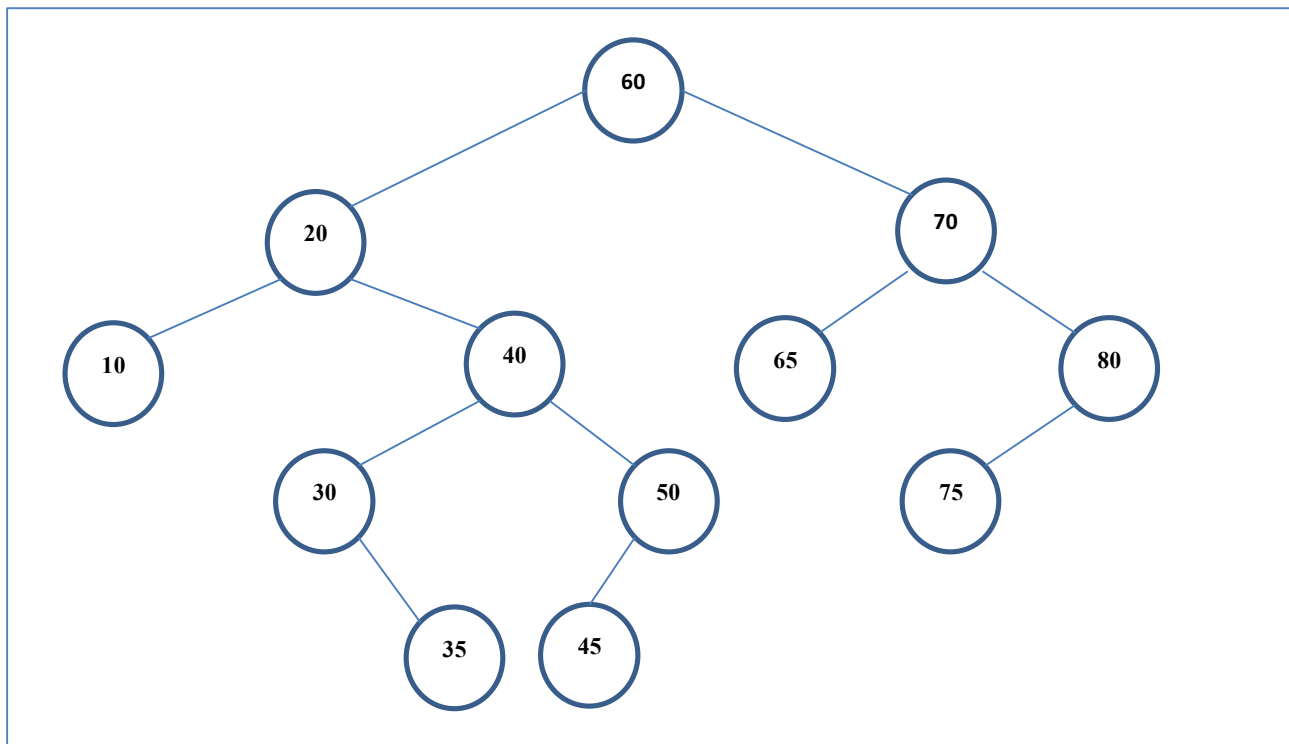
Step 3



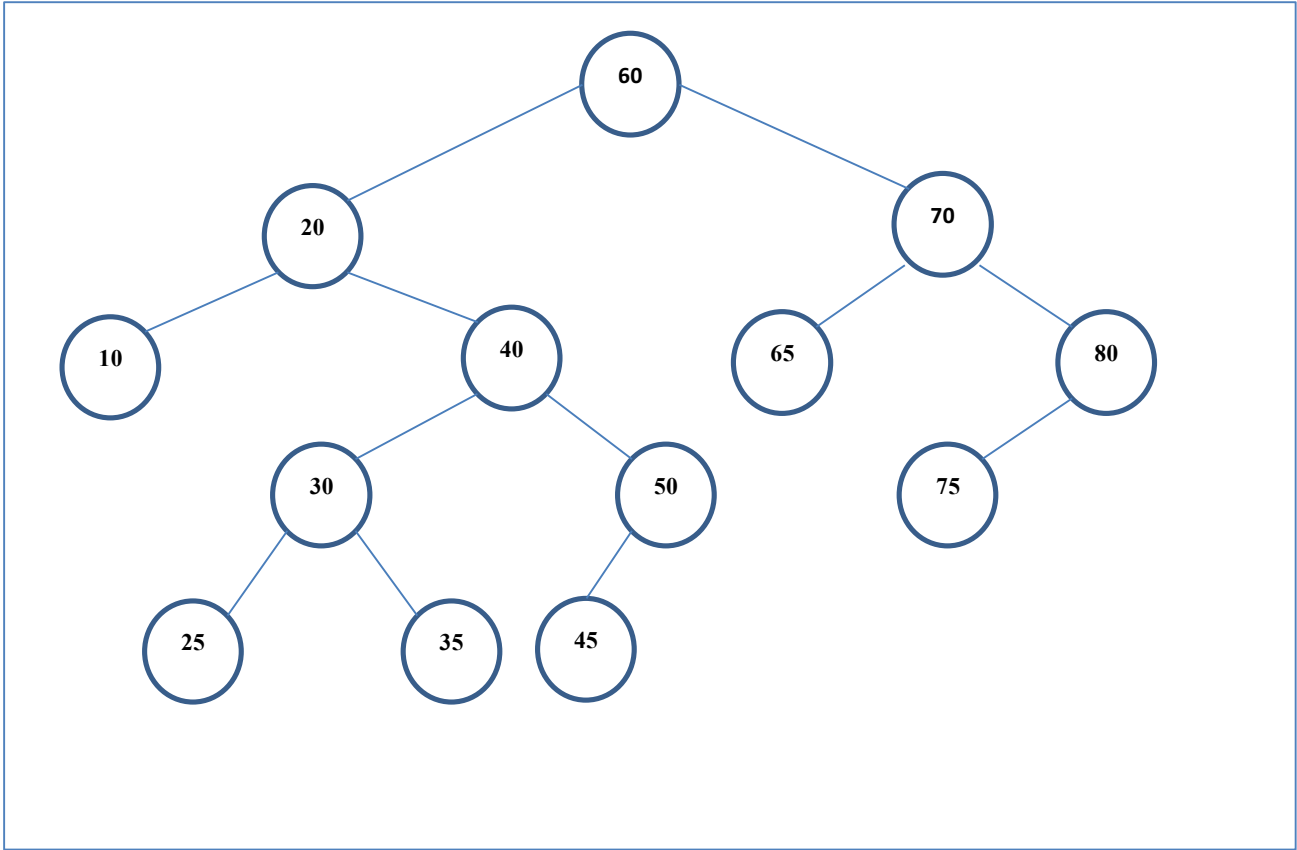
Step 4



Step 5

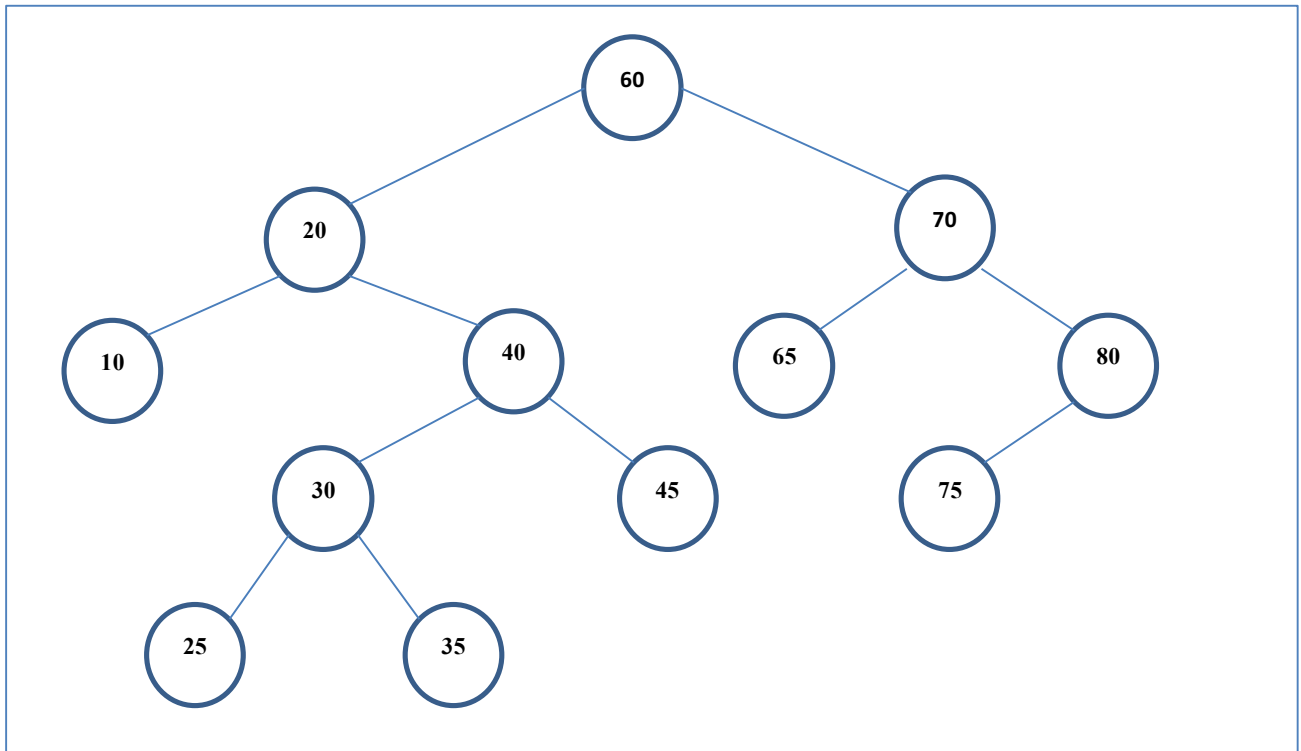


Step 6

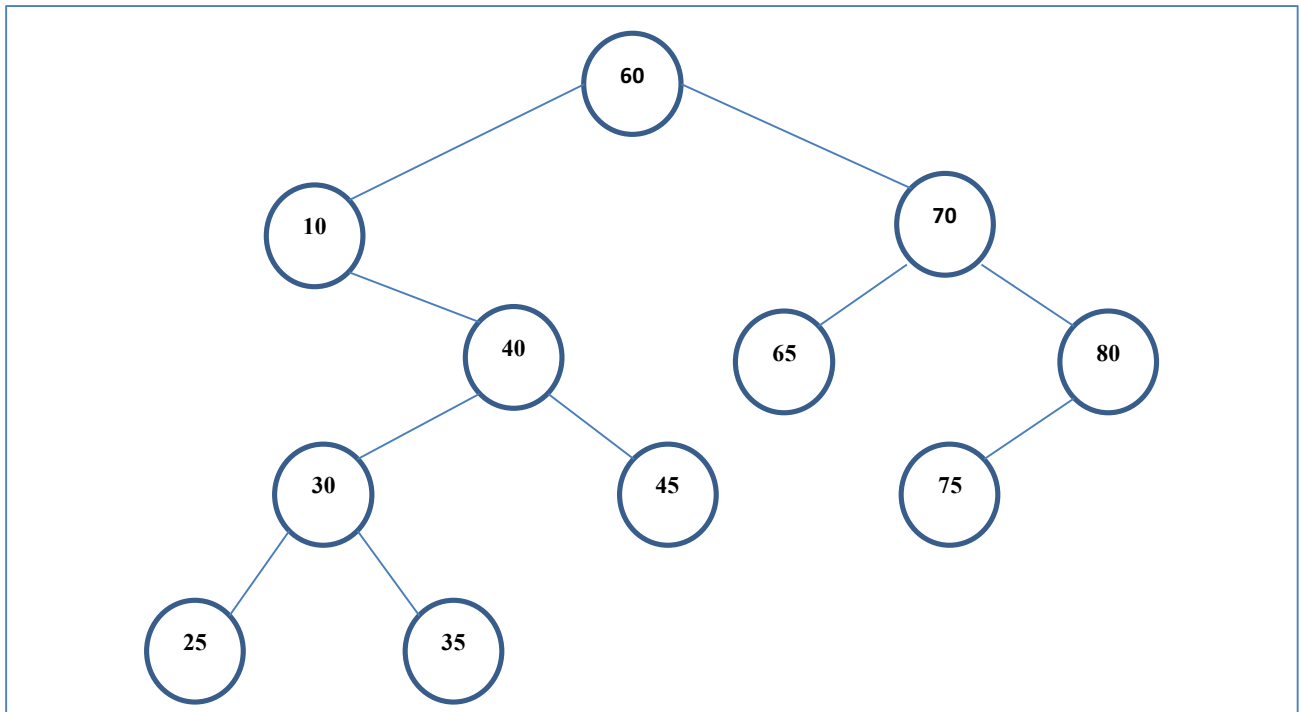


1. b.

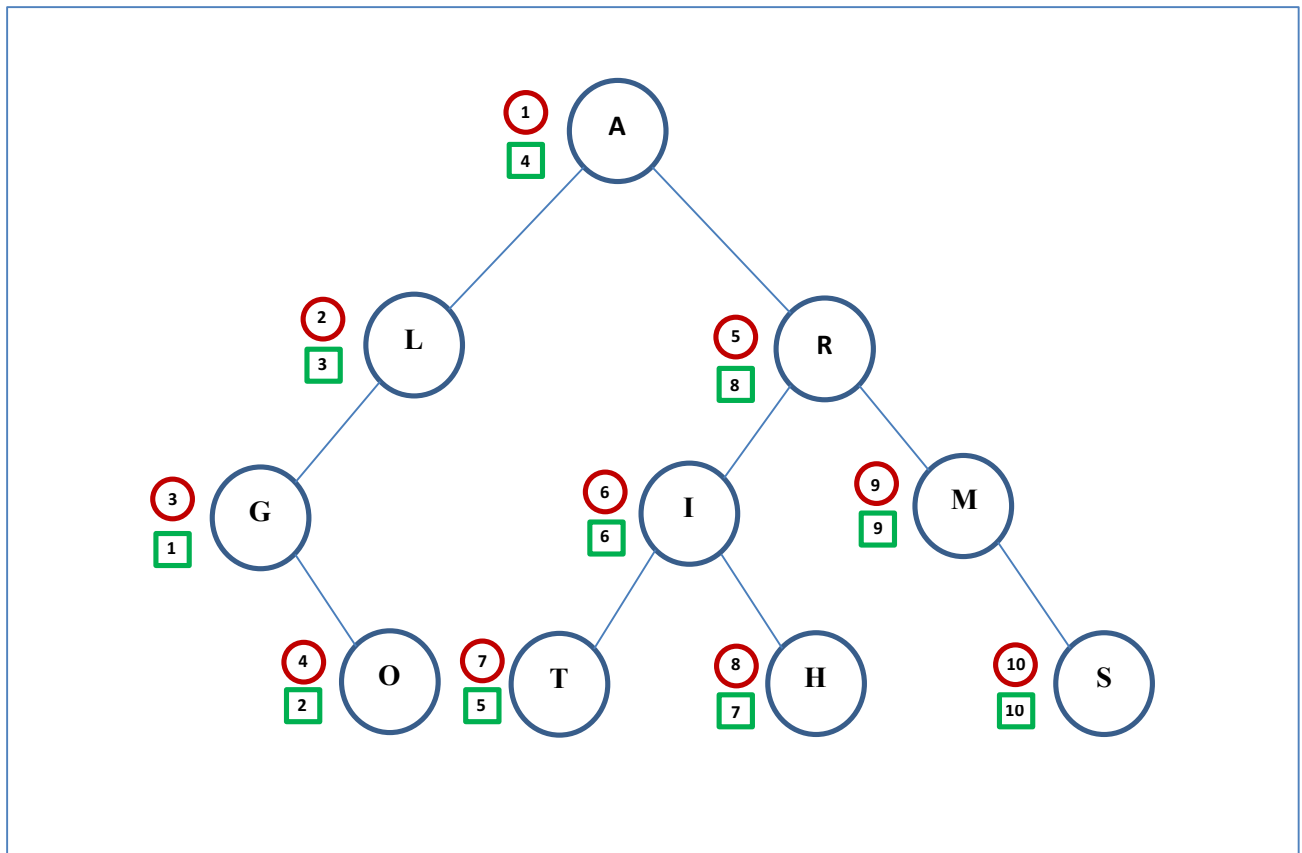
Step 1 (Deleting 50)



Step 2 (Deleting 20)



2.



○	Preorder Traversal	Each node is visited before its children.
□	In-order Traversal	The left subtree is visited first, then the node, then the right subtree.

3.

Situation	ADT
An alphabetic list of names.	Vector.
A grocery inventory ordered by the occurrence of the items in the store.	Priority queue.
The items on a cash register tape (with a dual tape: one is torn and given to the customer, the other is kept by the merchant. The one we are concerned with is the merchant's tape).	Priority queue.
A word processor that allows you to correct typing errors by using the backspace key.	Stack (Last In, First Out).
A collection of ideas in a chronological order.	Vector.
Air planes that stack above a busy airport, waiting to land.	Priority Queue.
People who are put on hold when they call a customer service number.	Queue (First In, First Out).
An employer who fires the most recently hired person.	Stack (Last In, First Out).

4.

Write a java program to count the number of elements in a singly linked list.

a. Iteratively b. Recursively

Please see Q4.java and relevant files in ./Q4 directory.

5.

Each element in a singly linked list L is an object with an attribute *key* and one pointer attribute *next*. Given an element x in the list, $x.next$ points to its successor in the linked list. If $x.next = NIL$, the element x has no successor and is therefore the last element or *tail*.

Code: Please see Q5.java and relevant files in ./Q5 directory

The following pseudocode illustrates a procedure to reverse a given singly linked list:

SINGLYLINKEDLISTREVERSED (L)

```
1   $x = L.head$ 
2   $previous = NULL$ 
3  while  $x.next \neq NULL$ 
4      // Reverse the link
5       $x.next = previous$ 
6       $previous = x$ 
7       $L.head = x$ 
```

As the procedure iterates through the entire list of n elements, it takes $O(n)$ time. The following is a Java implementation of the previous pseudocode:

6. Write a programs to compare sorting time using (a) selection sort and (b) quick sort to sort a list of records containing names. The file ds17s-asg2- data.txt contains some data to use for the sorting. Measure the time needed to perform the sort in both cases.

Please see Q6.java and relevant files in ./Q6 directory.