$\boxed{a}$ 1) $350-5(15-1) = 280$

2) Deleting a specified element in a singly linked list has a time complexity of O(n). To delete an element, we need to traverse the list until finding the node just before the target node Once identified, adjust the previous node's next pointer to skip the target node, effectively removing it from the list.

3) First ensure `P' and its successor (P. next) are not null. Update the next pointer of node P to point to the sucessor's next node: P.next = P.next.next; If the successor's next node is not null, update its previous pointer to skip the Successor and point back to P. if (p.next != null) P.next. prev = P;

4) Stacks and queues are both linear data structures that store elements sequentially. They can dynamically change in size as elements are added or removed. Also, access to elements is restricted. A stack follows a LIFO principle, a queue follows a FIFO principle.

5) CDEBA, ACDEB, EDCBA

6) Maximum number of nodes $= 2^0+2^1+2^2+2^3+2^4+2^5+2^6+2^7+2^8+2^9+2^{10} = 2047$

Minimum number of nodes $= 10+1 = 11$

7) Max nodes $= 2^{n+1} - 1$

$2^{n+1} - 1 = 128$

$2^{n+1} = 129$

$2^{7} = 128 \quad n = 7$

Depth of complete binary tree is 7

Leaf nodes $= 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 2^6 = 64$

$128 - 64 = 64$

64 leaf nodes in the complete binary tree

8) $C_4 = \dfrac{(2 \times 4)!}{(4+1)! \times 4!} = \dfrac{8!}{5! \times 4!} = 14$

14 different BSTs can be constructed with 4 distint keys

9) $N(0) = 1$

$N(1) = 2$

$N(2) = 1 + N(1) + N(0) = 4$

$N(3) = 1 + N(2) + N(1) = 7$

$N(4) = 1 + N(3) + N(2) = 12$

$N(5) = 1 + N(4) + N(3) = 20$

The minimum number of nodes needed for an AVL tree of heigh 5 is 20

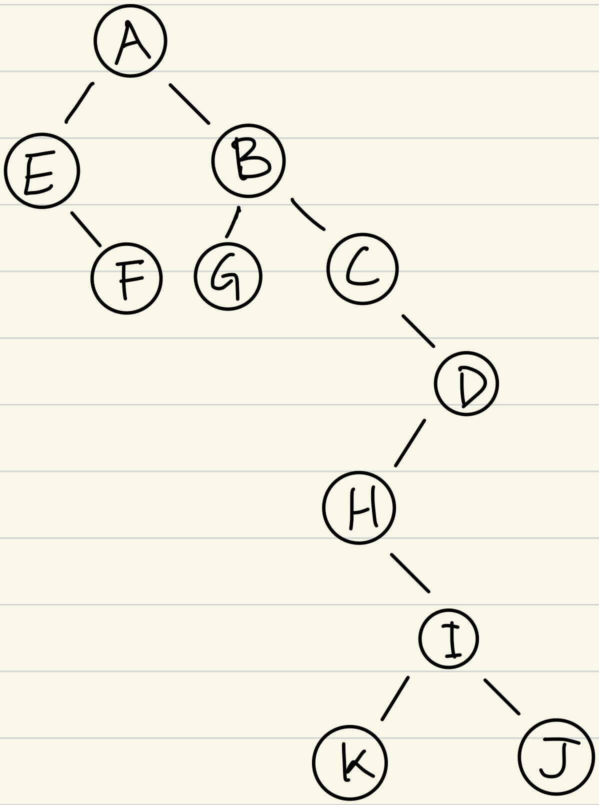
10) Binary Trees: A tree data structure where each node has at most two children, left and right child. Due to their hierarchical nature and Varied types, are versatile structures.

Balanced Binary Trees: A binary tree where the depth of the two subtrees of any node never differ by more than a specified margin.

AVL trees: A self balancing binary search tree where the difference betwee the heights of the left and right subtrees of any node is never more than one.

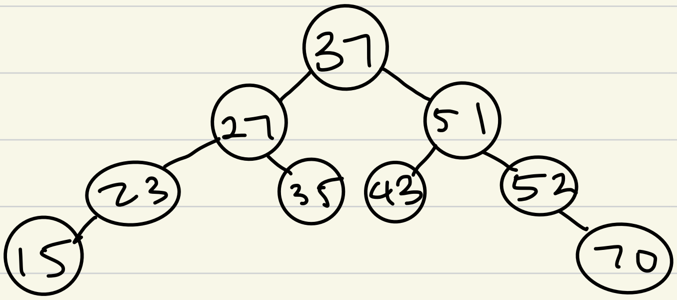
B trees. A balanced tree data structures that keep data sorted and allow for efficient insertion, deletion, and search operations.

$\boxed{6}$ a)

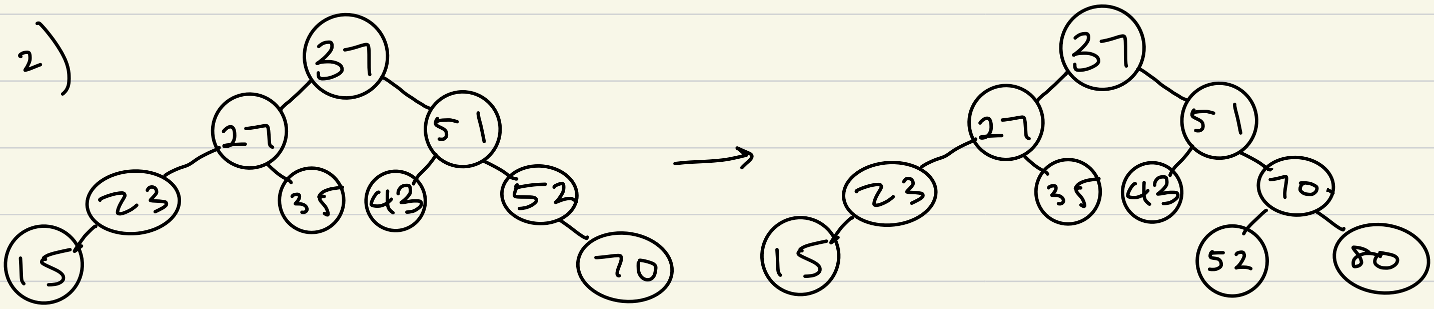


b) F E G K J I H D C B A

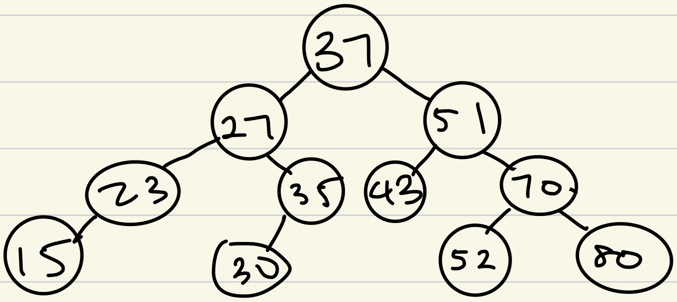
c 1)


37
27
51
23
35
43
52
15
70

2)


37
27
51
23
35
43
52
15
70
37
27
51
23
35
43
70
15
52
80

3)


37
27
51
23
35
43
70
15
30
52
80