Practice Questions : Disjoint Sets and Self balancing BST

1. Perform the following operations on disjoint set:
   a. Make-set
   b. Union
   c. Find-set

2. Perform the Union by-element_value (weight) operations on 10 elements (0-9, each initially in their own set). Draw the forest of trees that result U(1,5); U(3,7);U(1,4);U(5,7); U(0,8); U(6,9);U(3,9).
3. Perform union-by-rank for disjoint sets.
4. Perform path compression in tree-based disjoint sets. Verify using Find-set operation.
5. Find out the number of connected component in a given undirected graph and display their representative. You are free to choose representative in a given set. Vertices are numbered from 1 to V.
   Input: (T, $|V_i|$, Adj_i)
   2
   10
   0 1 1 0 0 0 0 0 0 0
   1 0 1 0 0 0 0 0 0 0
   1 1 0 1 0 0 0 0 0 0
   0 0 1 0 0 0 0 0 0 0
   0 0 0 0 0 1 1 0 0 0
   0 0 0 0 1 0 1 0 0 0
   0 0 0 0 1 1 0 0 0 0
   0 0 0 0 0 0 0 0 1 0
   0 0 0 0 0 0 0 1 0 0
   0 0 0 0 0 0 0 0 0 0
   10
   0 1 1 0 0 0 0 0 0 0
   1 0 1 0 0 0 0 0 0 0
   1 1 0 1 0 0 0 0 0 0
   0 0 1 0 0 0 0 0 0 0
   0 0 0 0 0 1 1 0 0 0
   0 0 0 0 1 0 1 1 0 0
   0 0 0 0 1 1 0 0 1 0
   0 0 0 0 0 1 0 0 1 0
   0 0 0 0 0 0 1 1 0 0
   0 0 0 0 0 0 0 0 0 0
   Output:
   4
   1 5 8 10
   3
   1 5 10

6. Check whether given graph is connected or not using disjoint sets.
   Input: (T, |Vi|, Adji)
   2
   6
   0 1 1 1 0 0
   1 0 1 0 1 0
   1 1 1 1 0 0
   0 1 0 1 0 0
   0 0 0 0 0 0
   6
   0 1 1 1 0 0
   1 0 1 0 1 1
   1 1 1 1 0 0
   0 1 0 1 0 0
   0 1 0 0 0 1
   Output:
   Disconnected
   Connected

7. Construct a binary search tree from given elements and perform the right rotation around root and print pre-order traversal.
   Input: (n, {x_i})
   7
   20 40 10 5 15 1 7
   Output:
   1 5 7 10 15 20 40
   10 5 1 7 20 15 40

8. Construct a binary search tree from given elements and perform the left rotation around root and print pre-order traversal.
   Input: (n, x_i)
   6
   50 69 90 99 57 31
   Output:
   31 50 57 69 90 99
   69 50 31 57 90 99
9. Construct an AVL search tree by inserting the following elements in the order of their occurrence. Print pre-order traversal.
   Input: (T, n_i, {x_i})
   2
   8
   64 1 14 26 13 110 98 85

6
10 20 30 40 50 25
Output:
14 1 13 64 26 98 85 110
30 20 10 25 40 50

10. Delete k elements from AVL tree. Print pre-order traversal.
    Input: (n, {x_i}, k, {x'_i})
    9
    9 5 10 0 6 11 -1 1 2
    5
    10 5 -1 6 11
    Output:
    9 1 0 -1 5 2 6 10 11
    1 0 -1 9 5 2 6 11
    1 0 -1 9 6 2 11
    6 1 0 2 9 11
    9 1 0 2 11
    1 0 9 2

11. Insert k elements into splay tree. Print pre-order traversal.
    Input: (T, n, {x_i})
    6
    100 50 200 40 30 20
    3
    25 55 35
    Output:
    25 20 50 30 40 100 200
    55 50 25 20 30 40 100 200
    35 30 25 20 40 50 55 100 200

12. Search k elements in splay tree. Print pre-order traversal.
    Input: (T, n_i, {x_i})
    6
    100 50 200 40 30 20
    3
    20 40 50
    Output:
    20 50 30 40 100 200
    40 30 20 50 100 200
    50 40 30 20 100 200