1: Space complexity is how much memory an algorithm needs based on input size. It includes:

Fixed space: Instructions, constants, and input-independent variables

Variable space: Input-dependent data structures and recursion stack

Return space: Memory for storing the output

2: In disjoint sets, Find locates which set an element belongs to by identifying its root.

```
FIND(x)
    if parent[x] != x
        parent[x] = FIND(parent[x]) // Path compression
    return parent[x]
```

Path compression flattens the tree by connecting all visited nodes directly to the root, making future operations faster.

3: Divide and Conquer breaks problems into smaller similar subproblems, solves them, and combines their solutions. The steps are:

Divide the problem into smaller parts

Solve each part separately

Combine solutions into a final answer