Recursive Generation of Unique Permutations

Q1-)You are asked to write program with a recursive algorithm for generating all unique permutations of an array with duplicate elements(You must use a recursive approach)

You need to generate all unique permutations of a given array. However:

1.The array may contain duplicate elements (e.g., [1, 1, 2]).
2.Identical elements in different orders should only be printed once.
3.Perform this operation recursively.
4.Use your own algorithm; do not use Python's built-in libraries (e.g., itertools.permutations).

Task: Write a recursive function:

•The function should take an array and the current permutation as parameters.
•Each permutation should be generated only once and printed.

Explanation of the Code

Input Parameters:

-arr: The input array containing elements (may include duplicates).
-current_perm: Tracks the current state of the permutation being generated.
-used: A list of boolean values indicating whether an element in arr has been used in the current recursion.

Base Case:

-If the length of current_perm equals the length of arr, print the permutation.

Example:

Input : [1, 1, 2]

Output:

[1, 1, 2]

[1, 2, 1]

[2, 1, 1]

Example2:

Input: [2, 2, 3, 3]

Output:

[2, 2, 3, 3]

[2, 3, 2, 3]

[2, 3, 3, 2]

[3, 2, 2, 3]

[3, 2, 3, 2]

[3, 3, 2, 2]

Example3:

Input:[1,2,3]

Output:

[1, 2, 3]

[1, 3, 2]

[2, 1, 3]

[2, 3, 1]

[3, 1, 2]

[3, 2, 1]