69. Given a collection of numbers, nums, that might contain duplicates, return *all possible unique permutations in any order.*

Example 1:

Input: nums = [1,1,2]

Output:

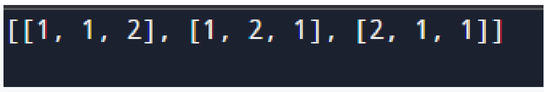
[[1,1,2], [1,2,1], [2,1,1]]

AIM : To find the permutations in any order

PROGRAM:

```
def permuteUnique(nums):
    def backtrack(path, counter):
        if len(path) == len(nums):
            result.append(path[:])
            return
        for num in counter:
            if counter[num] > 0:
                path.append(num)
                counter[num] -= 1
                backtrack(path, counter)
                path.pop()
                counter[num] += 1
    result = []
    counter = {}
    for num in nums:
        counter[num] = counter.get(num, 0) + 1
    backtrack([], counter)
    return result

nums = [1, 1, 2]
print(permuteUnique(nums))
```

OUTPUT: 

```
[[1, 1, 2], [1, 2, 1], [2, 1, 1]]
```

TIME COMPLEXITY: O(n!*n)