**Topic 6.10: Unique Permutations of an Array**

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

**Aim**  
To generate all unique permutations of an array that may contain duplicates using backtracking.

**Algorithm**

1. Sort the input array to group duplicates together.
2. Use backtracking with a visited array to track which elements have been used.
3. At each recursive call:
   * If the path length equals the array length, store the path as a valid permutation.
   * Skip over duplicate elements if they have been used in the same recursion level.
4. Continue until all unique permutations are generated.

**Output**A screenshot of a computer

AI-generated content may be incorrect.

**Result**  
The program generates all unique permutations of an array that may contain duplicates.

**Performance Analysis**

* Time Complexity: O(n × n!), since there are up to n! permutations and each takes O(n) to build.
* Space Complexity: O(n) for recursion depth and O(n!) for storing results.