Find All Subsets of Array with duplicate elements

LeetCode

This program focuses on generating subsets of a given array while handling duplicate elements. The objective is to create a set of all possible subsets where each subset can contain duplicate elements.

Approach 1: Function to find all subsets of 'nums' with duplicates using the backtracking approach

- Sort the input array to group duplicate elements together.
- Use a recursive backtracking function to generate subsets.
- At each index, decide whether to include or exclude the current element in the subset.
- Avoid duplicates by checking whether the current subset already exists in the result.
- **Time Complexity: O(2^n * n), where 'n' is the size of the input array due to the nested loops and duplicate checking.**
- **Space Complexity: O(2^n * n), considering the space required for the subsets and result.**

Approach 2: Function to find all subsets of 'nums' with duplicates using the backtracking approach (alternative implementation)

- Sort the input array to group duplicate elements together.
- Use a recursive backtracking function that skips duplicate elements.
- At each index, decide whether to include the current element in the subset, avoiding duplicates.
- **Time Complexity: O(2^n * n), where 'n' is the size of the input array due to the recursive function.**
- **Space Complexity: O(2^n * n), considering the space required for the subsets and result.**

Approach 3: Function to find the power set using the Bitwise approach

- Calculate the total number of subsets using bitwise left shift.
- Iterate through all possible subset combinations using bitwise operations.
- Construct subsets by including elements corresponding to set bits.
- Avoid duplicates by checking whether the current subset already exists in the result.
- **Time Complexity: O(2^n * n), where 'n' is the size of the input array due to nested loops and duplicate checking.**

- **Space Complexity: O(2^n * n), considering the space required for the subsets and result**