90. Subsets II

Given a collection of integers that might contain duplicates, nums, return all possible subsets (the power set).

Note: The solution set must not contain duplicate subsets.

Example:

```
Input: [1,2,2]
Output:
[
  [2],
  [1],
  [1,2,2],
  [2,2],
  [1,2],
  []
]
```

```
class Solution {
    public List<List<Integer>> subsetsWithDup(int[] nums) {

        // backtracking
        List<List<Integer>> results = new ArrayList<>();

        if (nums == null)
            return results;

        if (nums.length == 0) {
            results.add(new ArrayList<Integer>());
            return results;
        }

        Arrays.sort(nums); // contains duplicate
        List<Integer> subset = new ArrayList<>();
        helper(nums, 0, subset, results);
        return results;
    }

    private void helper(int[] nums, int start, List<Integer> subset,
                        List<List<Integer>> results) {
        results.add(new ArrayList<Integer>(subset));

        for (int i = start; i < nums.length; i++) {
            if (i != start && nums[i] == nums[i-1]) {
                continue;
            }

            subset.add(nums[i]);
            helper(nums, i+1, subset, results);
            subset.remove(subset.size()-1);
        }
    }
}
```