```
class Solution {
    public List<List<Integer>> subsets(int[] nums) {
        List<List<Integer>> res = new ArrayList<>();
        helper(res, new ArrayList<>(), nums, 0);
        return res;
    }

    private void helper(List<List<Integer>> res, List<Integer> tmp, int[] nums, int idx) {
        res.add(new ArrayList<>(tmp));
        for (int i = idx; i < nums.length; i++) {
            tmp.add(nums[i]);
            helper(res, tmp, nums, i+1);
            tmp.remove(tmp.size()-1);
        }
    }
}

class Solution {
    public List<List<Integer>> subsetsWithDup(int[] nums) {
        Arrays.sort(nums);
        List<List<Integer>> res = new ArrayList<>();
        helper(res, new ArrayList<>(), nums, 0);
        return res;
    }

    private void helper(List<List<Integer>> res, List<Integer> tmp, int[] nums, int idx) {
        res.add(new ArrayList<>(tmp));
        for (int i = idx; i < nums.length; i++) {
            if (i != idx && nums[i] == nums[i-1]) continue;
            tmp.add(nums[i]);
            helper(res, tmp, nums, i+1);
            tmp.remove(tmp.size()-1);
        }
    }
}
```