Subset Coding Pattern

LeetCode 78 - Subsets [*medium*]

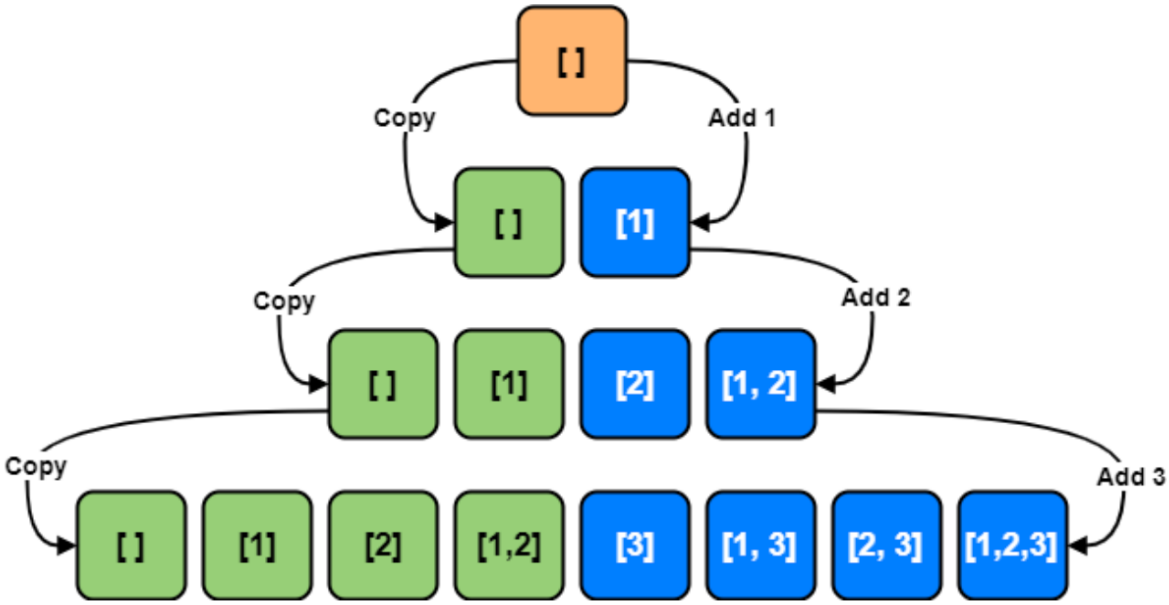
Given a set of *distinct* integers, `nums`, return all possible subsets (the power set).

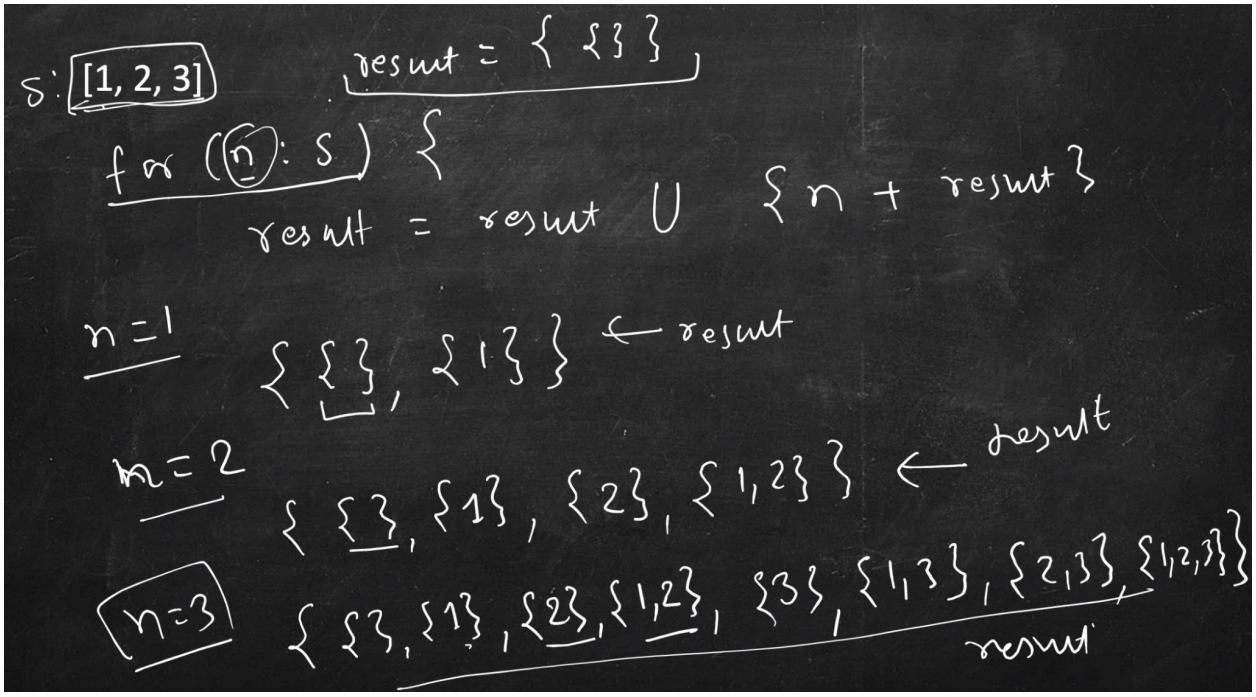
Note: The solution set must **not** contain *duplicate* subsets.

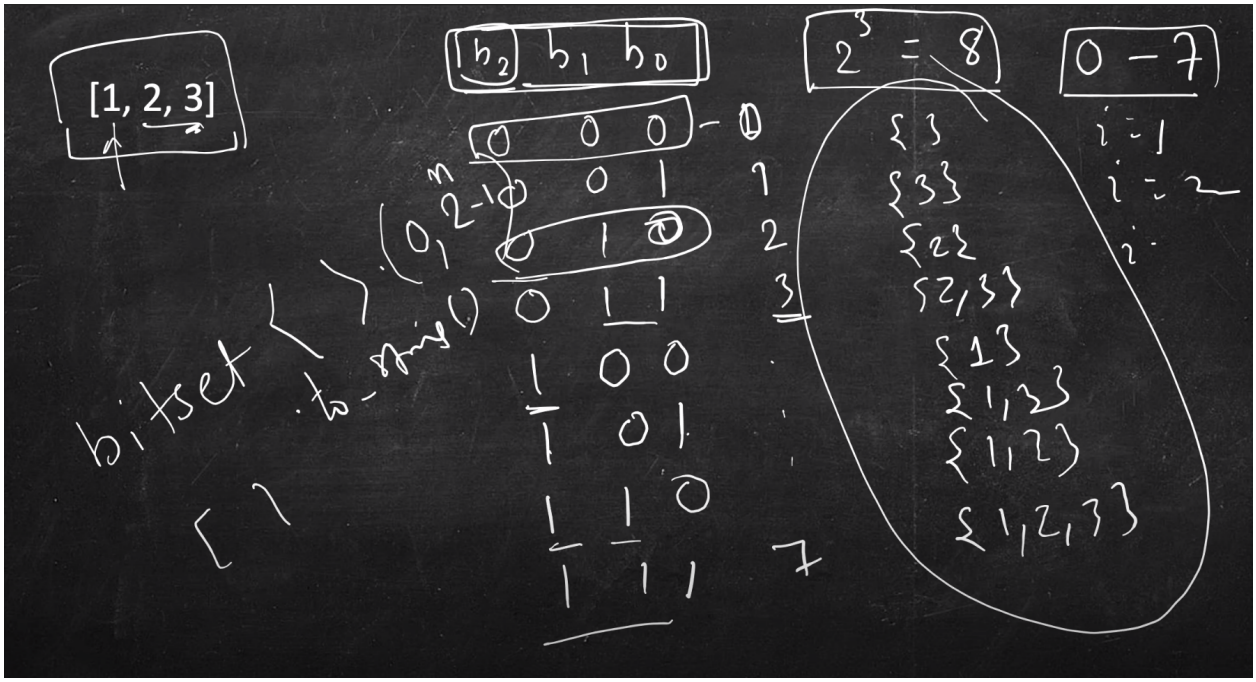
Example:

```
Input: nums = [1, 2, 3]
Output:
[
  [3],
  [1],
  [2],
  [1, 2, 3],
  [1, 3],
  [2, 3],
  [1, 2],
  []
]
```

1. Start with an empty set: **[[ ]]**
2. Add `num` (**1**) to existing sets: [[ ], **[1]**]
3. Add `num` (**2**) to existing sets: [[ ], [1], **[2], [1, 2]**]
4. Add `num` (**3**) to existing sets: [[ ], [1], [2], [1, 2], **[3], [1, 3], [2, 3], [1, 2, 3]**]




S: [1, 2, 3]
result = { {} }
for (n : S) {
result = result ∪ { n + result }
n = 1
{ {}, {1} } ← result
n = 2
{ {}, {1}, {2}, {1,2} } ← result
n = 3
{ {}, {1}, {2}, {1,2}, {3}, {1,3}, {2,3}, {1,2,3} }
result




```java
class Solution {
    public List<List<Integer>> subsets(int[] nums) {
        List<List<Integer>> result = new ArrayList();
        result.add(new ArrayList<Integer>());

        for(int x : nums){
            int n = result.size();
            for(int i = 0; i < n; ++ i){
                ArrayList<Integer> r = new ArrayList(result.get(i));
                r.add(x);
                result.add(r);
            }
        }

        return result;
    }
}
```

LeetCode 90 - Subsets II [*medium*]