

## 78. Subsets

13 February 2022 10:24 AM

Given an integer array `nums` of **unique** elements, return *all possible subsets (the power set)*.

The solution set **must not** contain duplicate subsets. Return the solution in **any order**.

### Example 1:

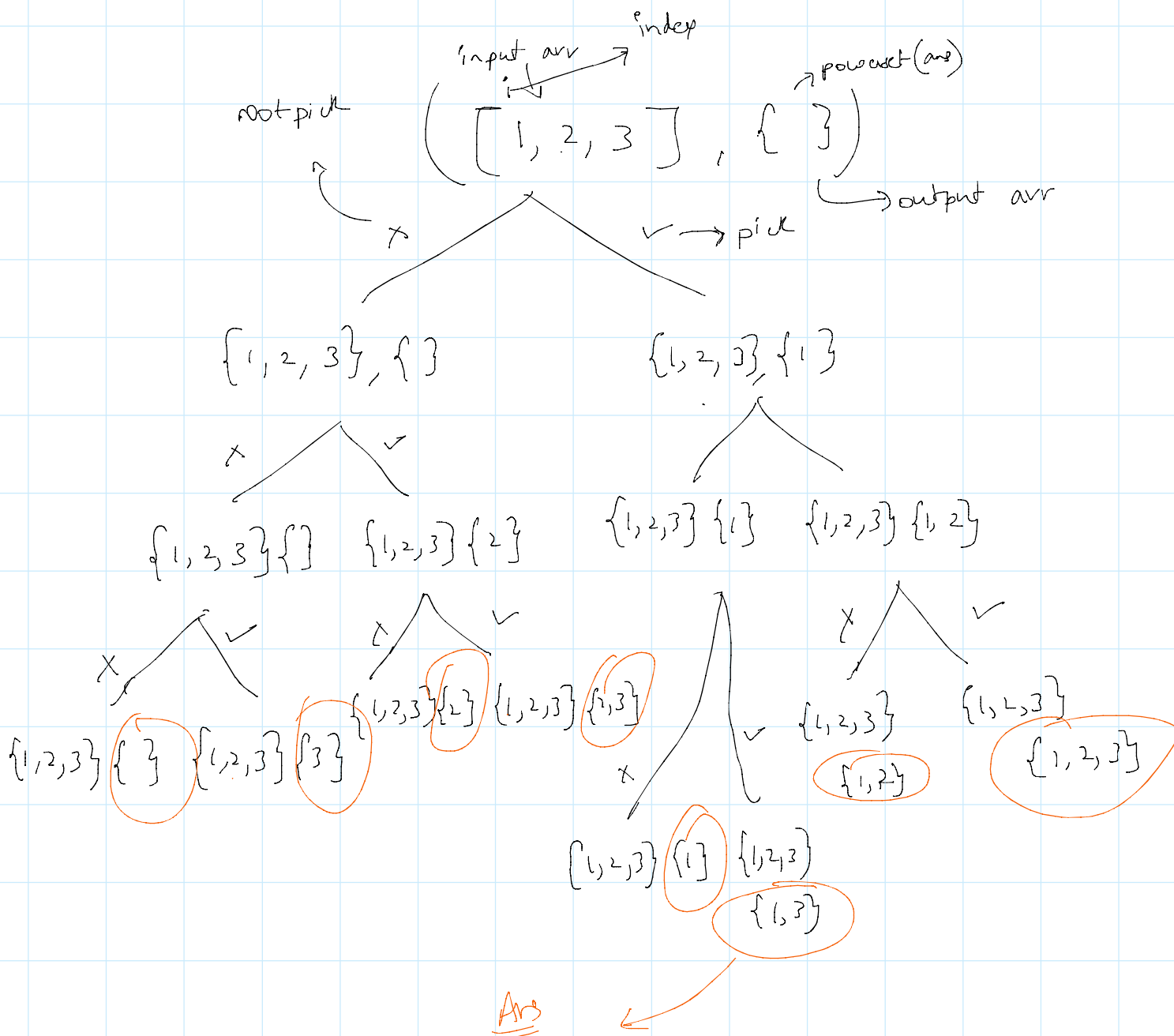
Input: `nums = [1,2,3]`

Output: `[[],[1],[2],[1,2],[3],[1,3],[2,3],[1,2,3]]`

### Example 2:

Input: `nums = [0]`

Output: `[[],[0]]`



```
class Solution {
private:
    void solve(vector<int> nums, vector<int> output, int index, vector<vector<int>> &ans){
        // base case
        if(index >= nums.size()){
            ans.push_back(output)
            return ;
        }

        //not pick
        solve(nums, output, index + 1, ans);

        // pick // include the elements
        int element = nums[index];
        output.push_back(element);
        solve(nums, output, index+1, ans)
    }
}

public:
    vector<vector<int>> subsets(vector<int>& nums) {

        vector<vector<int>> ans;
        vector<int> output;
        int index = 0;
        solve(nums, output, index, ans);
        return ans;
    }
};
```