

认识**本质之后**，这就是一道**模板题目**

第78题. 子集

题目地址：<https://leetcode-cn.com/problems/subsets/>

给定一组不含重复元素的整数数组 `nums`，返回该数组所有可能的子集（幂集）。

说明：解集不能包含重复的子集。

示例：输入：`nums = [1,2,3]`

输出：

```
[
  [3],
  [1],
  [2],
  [1,2,3],
  [1,3],
  [2,3],
  [1,2],
  []
]
```

题目可知该整数数组`nums`**不含重复元素组成**,

分析过程:

如果把组合问题、求子集问题、分割问题都抽象为一棵树的话，那么组合问题、分割问题都是收集从根节点到叶子节点的路径而子集问题则是求根节点到各个节点之间的路径

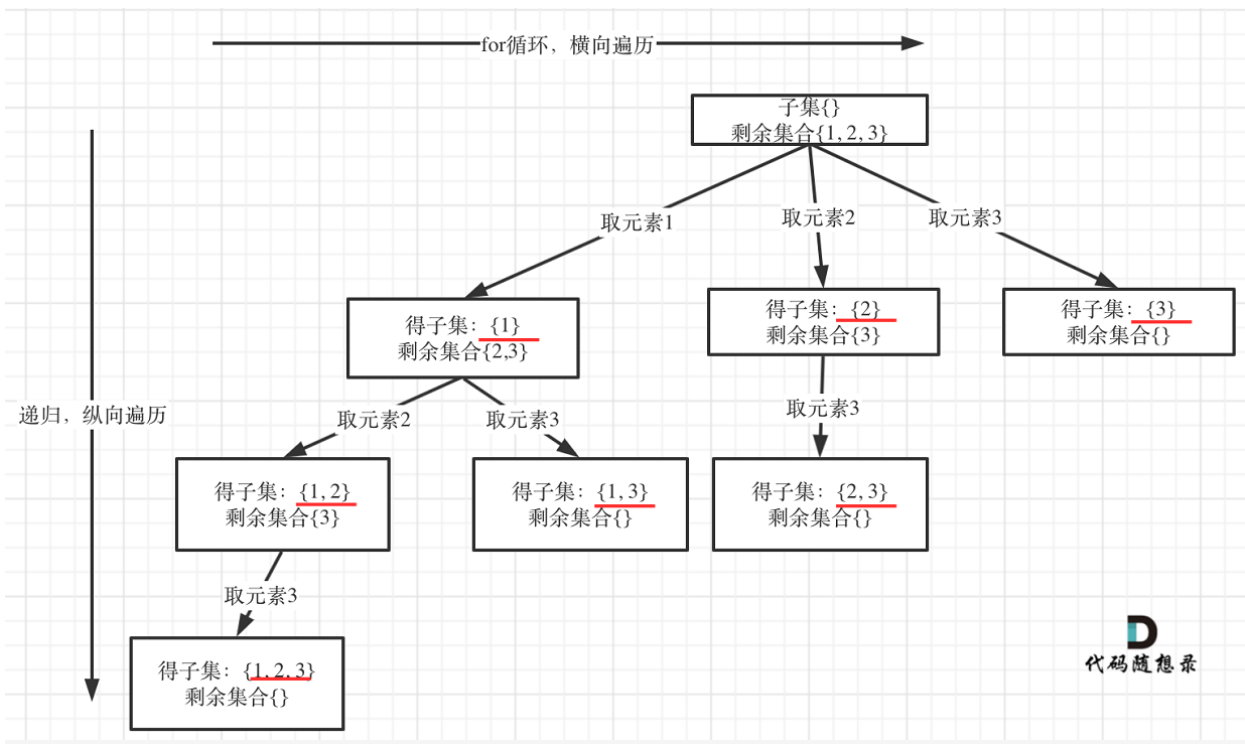
第二次强调:

回溯算法中**横向遍历**for循环的**起始位置从0开始还是从startIndex开始**

组合问题：从startIndex开始

排序问题：从0开始

求子集过程的树形结构:



已知当剩余集合为空的时候，就是叶子节点

那么**什么时候剩余集合为空**呢？

就是当startIndex已经大于数组的长度了，就终止了，因为**没有元素可取了**，

```

1 class Solution {
2 public:
3     vector<vector<int>> result;
4     vector<int> path;
5     void backtracking(vector<int>& nums, int startIndex){
6         result.push_back(path);
7         if(startIndex >= nums.size()){
8             return ;
9         }
10        for(int i=startIndex; i<nums.size(); i++){
11            path.push_back(nums[i]);
12            backtracking(nums, i+1);
13            path.pop_back();
14        }
15    }
16    vector<vector<int>> subsets(vector<int>& nums) {
17        backtracking(nums, 0);
18        return result;
19    }
20 };
  
```

总结：

子集问题，组合问题，分割问题的区别：

子集问题收集**树形结构中**树的**所有节点**的结果

组合问题和分割问题收集**树形结构中的叶子节点**的