

78 子集

Label: 回溯

给你一个整数数组 `nums`，数组中的元素 互不相同 。返回该数组所有可能的子集（幂集）。

解集 不能 包含重复的子集。你可以按 任意顺序 返回解集。

输入: `nums = [1,2,3]`

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

输入: `nums = [0]`

输出: `[[],[0]]`

- 迭代法枚举 (list)

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

        res.add(new ArrayList<>()); // 添加空集

        for (int i = 0; i < nums.length; i++) {
            int all = res.size();
            for (int j = 0; j < all; j++) {

                List<Integer> tmp = new ArrayList<>(res.get(j)); // 用 old list
                构造 new list
                tmp.add(nums[i]);
                res.add(tmp);
            }
        }
        return res;
    }
}
```

- 回溯

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

        List<List<Integer>> res = new ArrayList<>();
        backtrack(0, nums, res, new ArrayList<Integer>());
        return res;

    }

    private void backtrack(int i, int[] nums, List<List<Integer>> res,
        ArrayList<Integer> tmp) {
        // 走过的所有路径都是子集的一部分，都要加入到集合中
        res.add(new ArrayList<>(tmp));
        for (int j = i; j < nums.length; j++) {    // 控制条件隐含在了循环里
            // 做出选择
            tmp.add(nums[j]);
            // 递归
            backtrack(j + 1, nums, res, tmp);
            // 撤销选择
            tmp.remove(tmp.size() - 1);
        }
    }
}
```

- 位运算

```
class Solution {

    public static List<List<Integer>> subsets(int[] nums) {

        // 子集的长度是2的nums.length次方，这里通过移位计算
        int length = 1 << nums.length;

        List<List<Integer>> res = new ArrayList<>(length);

        // 遍历从0到length中间的所有数字，根据数字中1的位置来找子集
        for (int i = 0; i < length; i++) {
            List<Integer> list = new ArrayList<>();

            for (int j = 0; j < nums.length; j++) {
                // 如果数字i的某一个位置是1，就把数组中对应的数字添加到集合
                if (((i >> j) & 1) == 1)    // 对i进行移位操作，并取出最后一位来与1进行
                &, 从而判断i中的该位是否为1
                    list.add(nums[j]); // 注意，只有这里才用到了 num 数组中数字
            }

            res.add(list);
        }
        return res;
    }
}
```

