46 全排列

```
Label：回溯
给定一个不含重复数字的数组 nums ，返回其 所有可能的全排列 。你可以 按任意顺序 返回答案。
    输入：nums = [1,2,3]
    输出：[[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]

    输入：nums = [0,1]
    输出：[[0,1],[1,0]]

    输入：nums = [1]
    输出：[[1]]
```

- 回溯 （模板）

```
class Solution {
    public static List<List<Integer>> permute(int[] nums) {
        List<List<Integer>> re = new ArrayList<>();
        backtrack(re, new ArrayList<>(), nums,new boolean[nums.length]);// 回溯法的三要素，存储结果数组 re，临时结果(判断用的)，条件值nums。还有其他的额外控制条件
        return re;
    }

    private static void backtrack(List<List<Integer>> re, List<Integer> tempList, int [] nums, boolean[] mask) {

        if(tempList.size() == nums.length){
            re.add(new ArrayList<>(tempList));//重新拷贝结果 更改会改变这个list里面的值
        } else {
            // 回溯
            for (int i = 0; i < nums.length; i++) {
                if(mask[i] == true)  // 在这次中已经被使用过
                    continue;
                else {
                    tempList.add(nums[i]);
                    mask[i] = true;//访问过了就不要再访问
                    backtrack(re, tempList, nums, mask);
                    mask[i] = false;//一次回溯后就要将访问的重新置为未访问的
                    tempList.remove(tempList.size() - 1);  // 移出父节点，寻找下一个
                }
            }
        }
    }
}
```

> https://blog.csdn.net/tangyuan_sibal/article/details/90523192 回溯题模板