39 组合总数

Label：回溯
给你一个整数数组 nums 和一个整数 k ，请你返回其中出现频率前 k 高的元素。你可以按 任意顺序 返回答案。给定一个无重复元素的数组 candidates 和一个目标数 target ，找出 candidates 中所有可以使数字和为 target 的组合。
candidates 中的数字可以无限制重复被选取。
- 所有数字（包括 target）都是正整数。
- 解集不能包含重复的组合。

    输入：candidates = [2,3,5], target = 8,
    所求解集为：
    [
      [2,2,2,2],
      [2,3,3],
      [3,5]
    ]

- 回溯

```
class Solution {
    private List<List<Integer>> res = new ArrayList<>();

    public List<List<Integer>> combinationSum(int[] candidates, int target) {
        List<Integer> path = new ArrayList<>();
        Arrays.sort(candidates);
        backtrack(path,candidates,target,0,0);
        return res;
    }

    private void backtrack(List<Integer> path,int[] candidates,int target,int
sum,int begin) {
        if(sum == target) {
            res.add(new ArrayList<>(path));
            return;
        }

        for(int i = begin;i < candidates.length;i++) {  // 传入 begin，防止出现重复
结果
            int rs = candidates[i] + sum;
            if(rs <= target) {
                path.add(candidates[i]);
                backtrack(path,candidates,target,rs,i);
                path.remove(path.size()-1);
            } else {
                break;  // 排序后，可直接剪枝
            }
        }
    }
}
```