17 电话号码的字母组合

Label：回溯、深度遍历
给定一个仅包含数字 2-9 的字符串，返回所有它能表示的字母组合。答案可以按 任意顺序 返回。
给出数字到字母的映射如下（与电话按键相同）。注意 1 不对应任何字母。
输入：digits = "23" 输入：digits = "2"
输出：["ad","ae","af","bd","be","bf","cd","ce","cf"] 输出：["a","b","c"]

- 深度遍历

```
class Solution {
    Set<String> re = new HashSet<>();
    public List<String> letterCombinations(String digits) {
        HashMap<Integer, char[]> dict = new HashMap<>();
        dict.put(2,new char[]{'a','b','c'});
        dict.put(3,new char[]{'d','e','f'});
        dict.put(4,new char[]{'g','h','i'});
        dict.put(5,new char[]{'j','k','l'});
        dict.put(6,new char[]{'m','n','o'});
        dict.put(7,new char[]{'p','q','r','s'});
        dict.put(8,new char[]{'t','u','v'});
        dict.put(9,new char[]{'w','x','y','z'});
        deep(digits,dict);
        return new ArrayList(re);
    }
    private void deep(String reStr, HashMap<Integer, char[]> dict) {    // 用reStr代替index递增
        if (reStr == null || reStr.length() == 0)  return;
        else {  // 拼接成新串，传入set
            Integer i = Integer.valueOf(reStr.substring(0, 1));
            char[] chars = dict.get(i);
            if (re.size() == 0) {
                for (Character c : chars) re.add(c.toString());
            }else {
                Set<String> newSet = new HashSet<String>();
                for (String str : re) {
                    for (Character c : chars) newSet.add(str + c.toString());
                }
                re = newSet;
            }
            deep(reStr.substring(1, reStr.length()), dict);  // 继续
        }
    }
}
```