LeetCode 题库分门别类详细解析笔记（第8章）

第8章 递归和回溯法 3

8-1 在树形问题中使用递归（1题） 3

 例题1：LeetCode 第 17 题 3

8-2 回溯法是暴力解法的一个主要的实现手段（2题） 4

 练习1：LeetCode 第 93 题 4

 练习2：LeetCode 第 131 题 4

8-3 排列问题 Permutations（3题） 4

 例题1：LeetCode 第 46 题 5

 练习1：LeetCode 第 47 题 5

 练习2：排列这个问题还有什么更好的方法。 5

8-4 组合问题 Combinations（1题） 5

 例题1：LeetCode 第 77 题 5

8-5 回溯法解决组合问题的优化（6题） 6

 例题1：上一节的例子。 6

 练习1：LeetCode 第 39 题 6

 练习2：LeetCode 第 40 题 6

 练习3：LeetCode 第 216 题 6

 练习4：LeetCode 第 78 题 7

 练习5：LeetCode 第 90 题 7

 练习6：LeetCode 第 401 题 7

8-6 二维平面上使用回溯法（1题） 7

 例题1：LeetCode 第 79 题 7

8-7 floodfill 算法，一类经典问题 Number of Islands（3题） 8

 例题1：LeetCode 第 200 题 8

 练习1：LeetCode 第 130 题 8

 练习2：LeetCode 第 417 题 8

8-8 回溯法是经典的人工智能的基础（3题） 8

 例题1：LeetCode 第 51 题 9

 练习1：LeetCode 第 52 题 9

 练习2：LeetCode 第 37 题 9

模板：

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

第8章 递归和回溯法（20题）

回溯法是解决很多算法问题的常见思想，甚至可以说是传统人工智能的基础方法。其本质依然是使用递归的方法在树形空间中寻找解。在这一章，我们来具体看一下将递归这种技术使用在非二叉树的结构中，从而认识回溯这一基础算法思想。

8-1 在树形问题中使用递归（1题）

* 例题1：LeetCode 第 17 题

题目要求：电话号码的字母组合。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。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/letter-combinations-of-a-phone-number/description/

英文网址：

求解关键：

对于问题的考虑：

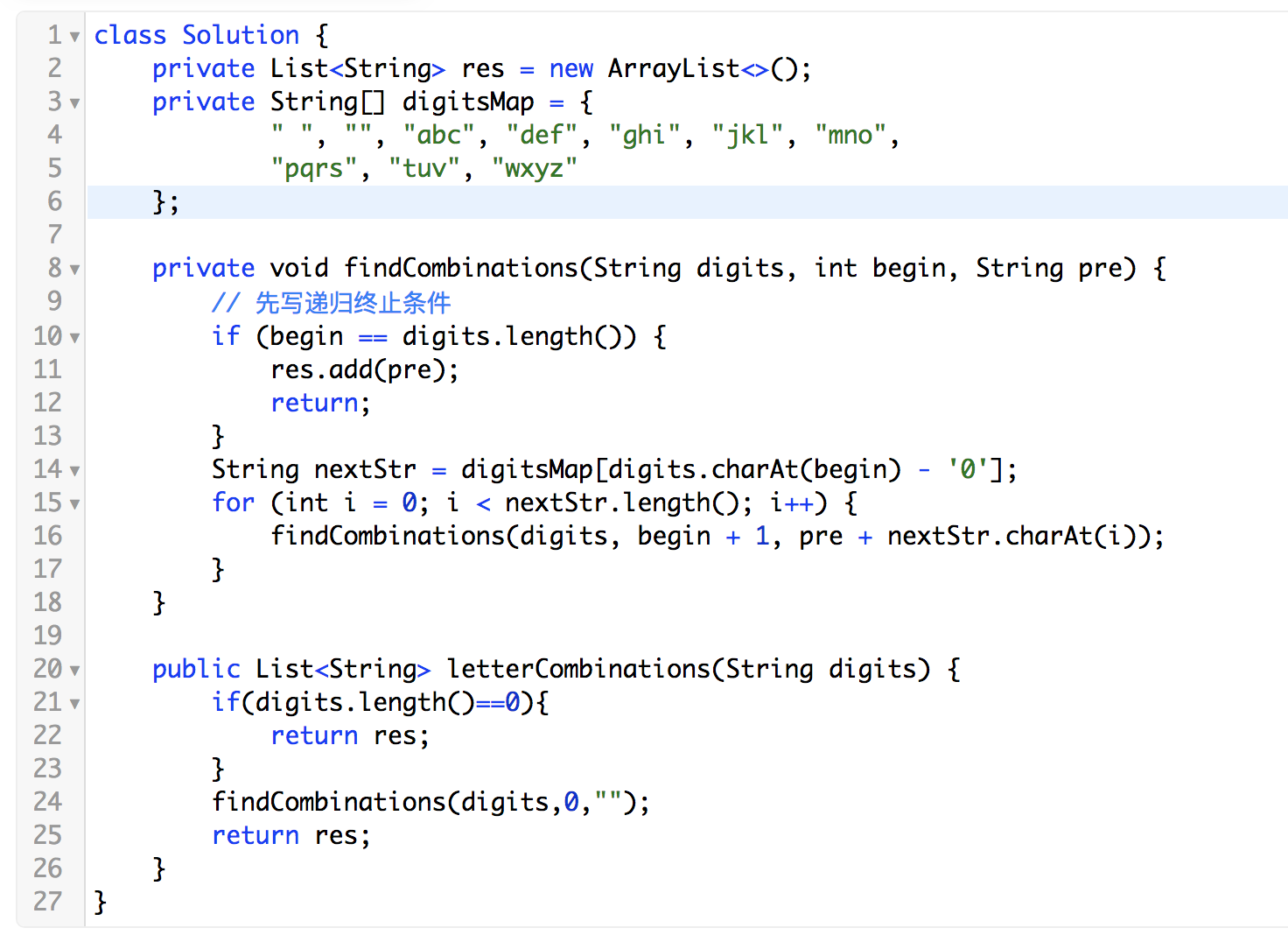
1、字符串的合法性（能出现什么样的字符串，“@”可以吗？“1”可以吗？）：不可以，只能出现 2、3、4、5、6、7、8、9。

2、空字符串（如果给出的是空字符串，返回什么）：返回空列表，即元素个数为 0 的列表。

3、多个解的顺序？题目已经告诉了我们，这个问题中对顺序没有要求。

我认为，这个问题更像数学问题中的乘法计数原理：第 1 步，考虑数字 2 能表达的三个字母；第 2 步，考虑 3 能表达的三个字母。于是，我们最容易想到用多重循环来解决这个问题。但是在步数很多的时候，循环就变得低效了。此时，递归回溯这个技巧就可以派上用场了。

我的解答：https://gist.github.com/liweiwei1419/dce4c453e48fafd81f45afe3531570b0



参考资料：

思考总结：

1、为什么我们要再写一个函数，而不是直接在原来的 letterCombinations 函数中书写呢？递归过程是把一个规模较大的问题一步一步地转化成为一个规模更小的问题，而我们发现的递归关系并不能用 letterCombinations 函数来描述，也就是说这个规模更小的问题，不能使用 letterCombinations() 来表述；

2、细节考虑要周到：我们在使用递归方法解决问题的时候，一定不能忽略边界的情况的处理；同时递归终止条件要仔细考虑，特别是对边界的情况；

3、如何设计递归方法其实是有固定模式的，参数的设定也是有规律的，无非是弄清楚之前是什么，当前是什么，然后把当前的加到之前的；

4、对于数字字符转换为数字，这里使用的是 `digits.charAt(index) - '0'`；

5、严格意义上说，还要对所输入的数字字符的合法性作判断，例如：`assert c >= '0' || c <= '9' || c != '1';`；

6、findCombination 函数中的 digitsMap 可以写成成员变量；

7、这里因为 String 是不可变对象，所以每一次的方法调用，其实都是新的对象传递下去，这一点在我们后续的练习中要留意（这句话表达比较隐晦，要深刻理解这个事实还要做后面的练习，当 result 是其它类型的对象的时候，就不能简单的 add 操作了）。

8、可以看到，递归回溯的结果是很整齐的，在后序的学习中我们就会看到，递归回溯是深度优先遍历的一种体现。

8-2 回溯法是暴力解法的一个主要的实现手段（2题）

对上节的例题添加打印输出，观察执行路径。

* 练习1：LeetCode 第 93 题

题目要求：给定一个只包含数字的字符串，复原它并返回所有可能的 IP 地址格式。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/restore-ip-addresses/description/

英文网址：

求解关键：

1、使用深度优先遍历、递归回溯的思想来完成；

2、IP 地址一共 4 段，每一段的最大值是 255，最小值是 0，我们采用搜索的办法来得到有效的 ip 段；

3、每一次循环判断接下来读进来的 3 个数字字符是有可能成为一个 ip 段，如果可以，加到已经形成的 ip 段后面（特别要注意，截取字符串的时候不能越界）；

4、接下来递归终止的条件就得分析清楚了，但是也不是难事，把握好总共分 4 段，当画上第 4 个点，并且下一个考察的下标已经“刚刚好”越界的时候，此时，我们就找到了一个有效的 ip 段分割。

我的解答：https://gist.github.com/liweiwei1419/ddbc323032be47a62c57cdecb4c9df7a

参考资料：

思考总结：明白了这些问题的套路以后，题目的难度就变成了“**简单**”。

* 练习2：LeetCode 第 131 题

题目要求：分割回文串。给定一个字符串 s，将 s 分割成一些子串，使每个子串都是回文串。返回 s 所有可能的分割方案。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/palindrome-partitioning/description/

英文网址：

求解关键：

我的解答：https://gist.github.com/liweiwei1419/da50f8c99787a4a20bfe63ab48f69aed

参考资料：

思考总结：

8-3 排列问题 Permutations（3题）

* 例题1：LeetCode 第 46 题

题目要求：给定一个没有重复数字的序列，返回其所有可能的全排列。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/permutations/description/

英文网址：

求解关键：

1、画图理解思路是关键；

2、打印出一些信息观察程序的执行流程。

我的解答：https://gist.github.com/liweiwei1419/d635a147e34d8a86fe4258426a9161df

参考资料：

思考总结：多刷几次也基本知道套路了，所以多实践很重要，实践有助于对问题的思考，而并非一定一直看笔记。

* 练习1：LeetCode 第 47 题

题目要求：

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/permutations-ii/description/

英文网址：

求解关键：找到重复的原因，对树进行剪枝。

1、首先将数组排序；

2、只处理第 1 次遇到的那个数，举个距离的例子画个图。重点理解：i>0 并且 nums[i]==nums[i-1] 并且之前那个数已经处理完毕（回溯完成）!used[i-1]。

我的解答：https://gist.github.com/liweiwei1419/0a794dce38f70d973e1a2724a301a033

参考资料：

思考总结：

* 练习2：排列这个问题还有什么更好的方法。

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

8-4 组合问题 Combinations（1题）

* 例题1：LeetCode 第 77 题

题目要求：给定两个整数 n 和 k，返回 1 ... n 中所有可能的 k 个数的组合。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/combinations/description/

英文网址：

求解关键：按顺序查找，已经用过的数字就不会再使用，因此不用设置 used 数组。

我的解答：https://gist.github.com/liweiwei1419/1c2704e0134685bd9ddfacc3671f008d

参考资料：

思考总结：

8-5 回溯法解决组合问题的优化（6题）

* 例题1：上一节的例子。

题目要求：给定两个整数 n 和 k，返回 1 ... n 中所有可能的 k 个数的组合。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/combinations/description/

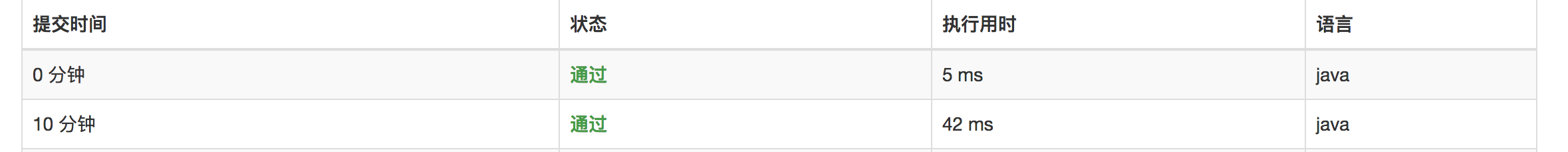
英文网址：

求解关键：重点分析： n - (k - stack.size()) + 1

我的解答：https://gist.github.com/liweiwei1419/1c2704e0134685bd9ddfacc3671f008d

参考资料：

思考总结：正确的键值可以极大提高算法的效率。



* 练习1：LeetCode 第 39 题

题目要求：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。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/combination-sum/description/

英文网址：

求解关键：

（1）这道题猛地一看好像跟前面的问题搭不上关系，因为题目中说“candidates 中的数字可以无限制重复被选取”。（2）但其实仔细想想就会发现，我们每次取数字的时候，还从原点开始取就行了呀，是不是很酷。

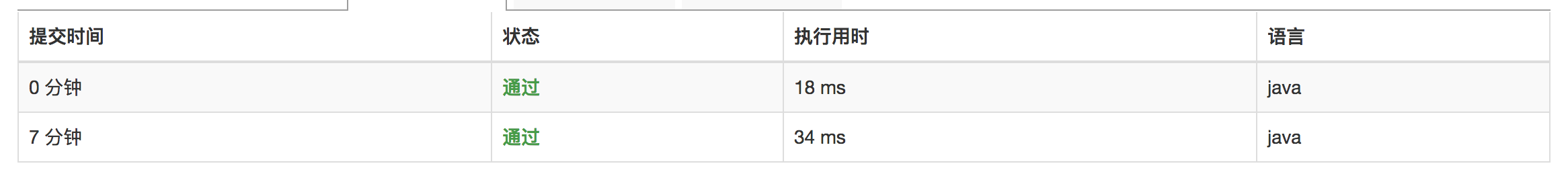
（3）为了达到提前判断循环结束的效果，我们可以先对数组排个序，如果起点数字比剩下的和还要大，后面的循环就没有必要进行下去了。此时，我们在 for 循环里加判断，尽量减少了系统栈的调用深度。

我的解答：https://gist.github.com/liweiwei1419/c13a7817dcce220311b51969a01759ad

参考资料：

思考总结：

优化的版本比之前节约了近一半的时间。



* 练习2：LeetCode 第 40 题

题目要求：

题目难度：

中文网址：https://leetcode-cn.com/problems/combination-sum-ii/description/

英文网址：

求解关键：

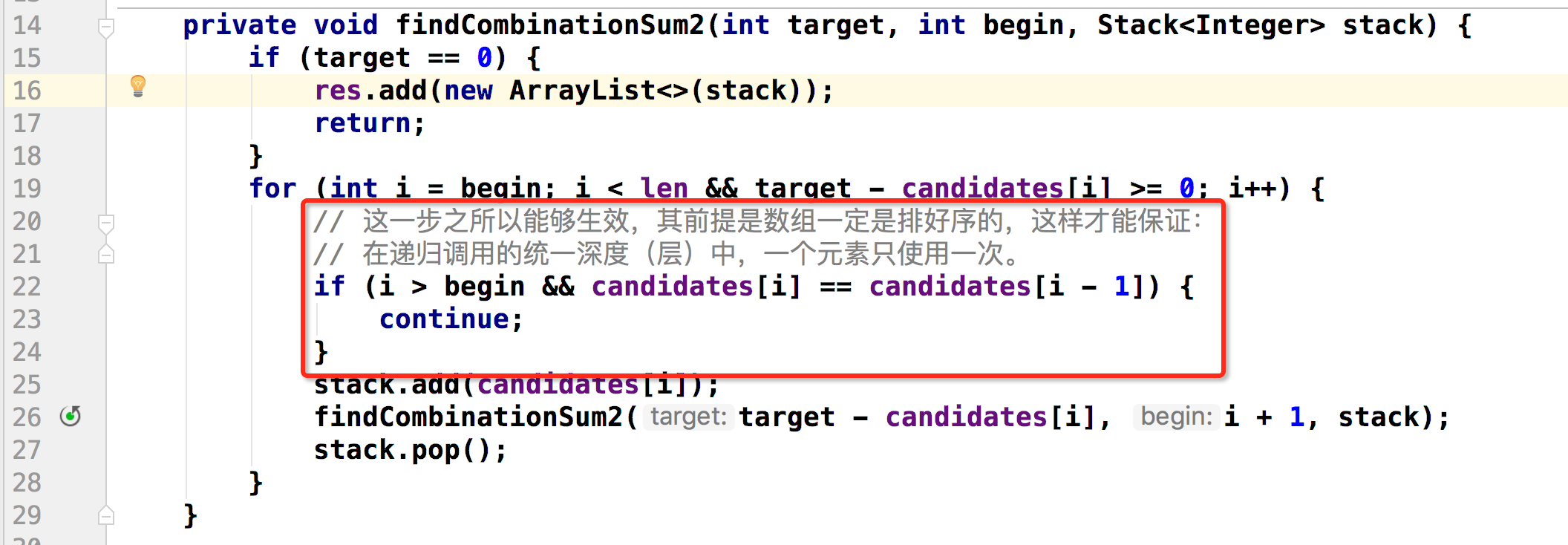
（1）与上一题的区别，39 题的数组没有重复数字，可以使用多次；40 题的数组有重复数字，只能使用一次。

（2）很容易想到，应该先对给出的数组进行排序，排序的目的有两个：其一是，可以提前终止循环，其二是“**在递归函数的调用中，统一深度的一个值只能使用一次**”，**这一处理也几乎成为了标准写法，或许刚刚开始接触的时候并不好理解，应该使用具体的例子画出图来理解，然后多做一些类似练习，理解代码为什么那样写**。

我的解答：https://gist.github.com/liweiwei1419/c1acc2f41cfa772e7f5a1bf298b9f9bb

参考资料：

思考总结：



未使用去重的结果：[[1, 1, 6], [1, 2, 5], [1, 7], **[1, 2, 5]**, **[1, 7]**, [2, 6]]

使用了去重的结果：[[1, 1, 6], [1, 2, 5], [1, 7], [2, 6]]

* 练习3：LeetCode 第 216 题

题目要求：找出所有相加之和为 n 的 k 个数的组合。组合中只允许含有1 - 9的正整数，并且每种组合中不存在重复的数字。

题目难度：**中等**。

中文网址：https://leetcode-cn.com/problems/combination-sum-iii/description/

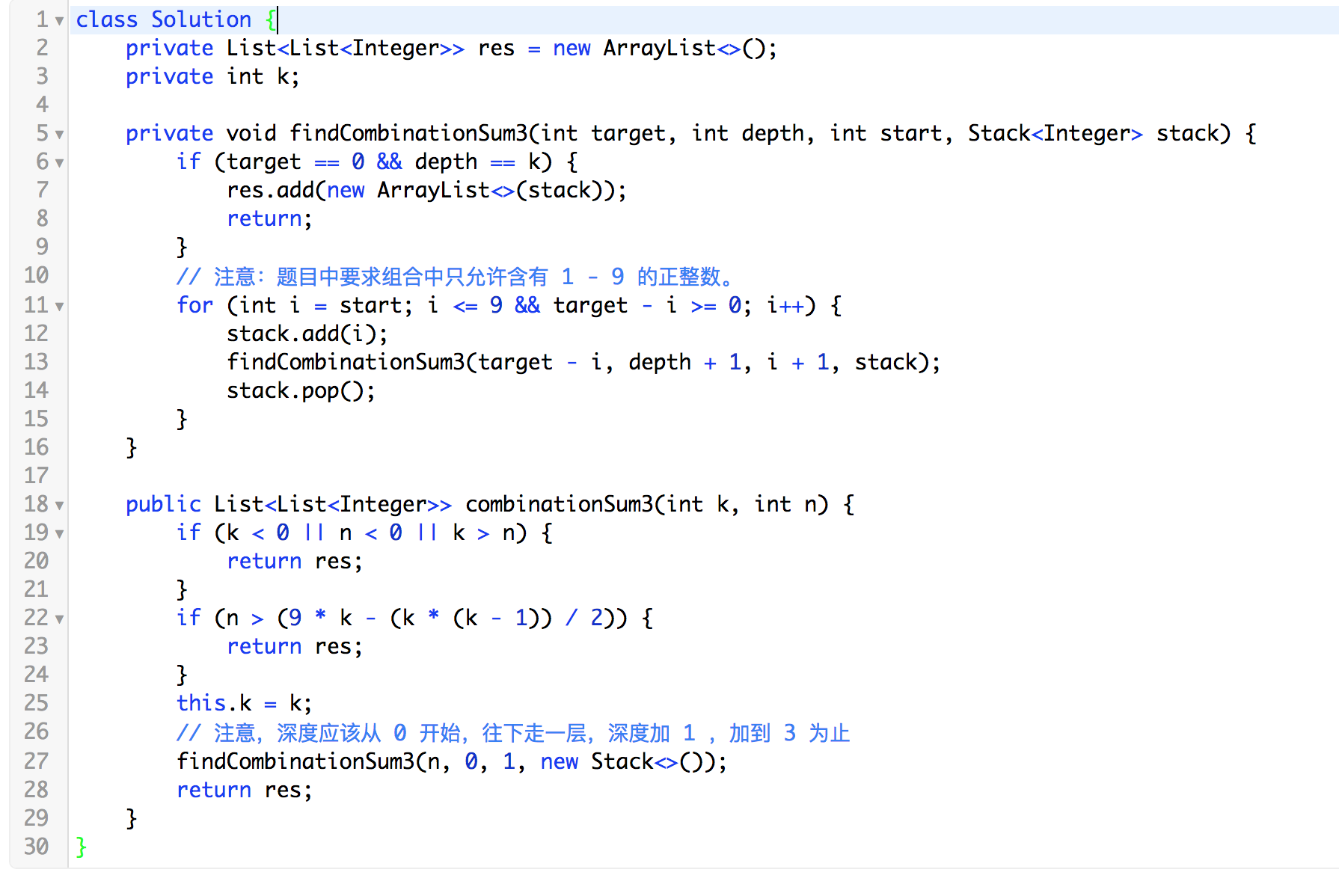
英文网址：

求解关键：其实写到这里，有些步骤已经非常熟悉了，基本不用动脑子就能写出来，因此比较容易误入歧途，这里说的歧途是不能很好理解题意，不能很好区分我们已经做过的题和新题的区别，哪些可以照搬，哪些是要特别处理的。对于这道题来说，要注意的是：

（1）“组合中只允许含有1 - 9的正整数”，因此，其实 n 其实是有一个上限的，n 不能无限大，这个上限通过归纳，通过计算一个等差数列的前 k 项和得到；

（2）还是因为“组合中只允许含有1 - 9的正整数”，所以，递归调用到下层的时候，最终值最大只能是 9。

我的解答：https://gist.github.com/liweiwei1419/7f0a16cd165b2ae3796a563bf9bca6f2



参考资料：

思考总结：

* 练习4：LeetCode 第 78 题

题目要求：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）。

题目难度：**中等**。

中文网址：https://leetcode-cn.com/problems/subsets/description/

英文网址：

求解关键：这道题有了前面求解组合数的基础，就不是什么难事了，在之前的基础上做一层循环，就可以得到一个 AC 解。

我的解答：https://gist.github.com/liweiwei1419/379a168369b67ab10791af40c35f0c1a

参考资料：

思考总结：虽然代码编写正确，但是具体的执行流程还不是特别清晰。我的理解是，沿途（深度优先遍历的路径），就把每一步的结果记录下来。所以改动的地方其实很小：把递归终止条件去掉即可。

* 练习5：LeetCode 第 90 题

题目要求：

题目难度：

中文网址：https://leetcode-cn.com/problems/subsets-ii/description/

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：虽然代码编写正确，但是具体的执行流程还不是特别清晰。我的理解是，沿途（深度优先遍历的路径），就把每一步的结果记录下来。所以改动的地方其实很小：把递归终止条件去掉即可。

* 练习6：LeetCode 第 401 题

提示：是一个典型的组合问题

题目要求：

题目难度：简单。

中文网址：https://leetcode-cn.com/problems/binary-watch/description/

英文网址：

求解关键：其实一点都不难，就是有点繁琐。

我的解答：

参考资料：

思考总结：

（1）这道理递归终止条件跟以前不一样，一种方式是 ++ ，加到够数为止，一种方式是 -- ，减到 0 为止；（2）如果传递的是值，回溯的时候就不用把状态重置。

8-6 二维平面上使用回溯法（1题）

* 例题1：LeetCode 第 79 题

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

8-7 floodfill 算法，一类经典问题 Number of Islands（3题）

* 例题1：LeetCode 第 200 题

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

* 练习1：LeetCode 第 130 题

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

* 练习2：LeetCode 第 417 题

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

8-8 回溯法是经典的人工智能的基础（3题）

* 例题1：LeetCode 第 51 题

n 皇后问题，典型的递归回溯问题。

n 皇后问题有很多优化的思路：加快搜索的过程。

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

* 练习1：LeetCode 第 52 题

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：

* 练习2：LeetCode 第 37 题

提示：求解数独（这是比 n 皇后问题更酷的问题，典型的人工智能的问题，自动来解决，递归加上回溯，有效剪枝，人工智能的开始章节一般就将搜索问题）

题目要求：

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

思考总结：