LeetCode 题库分门别类详细解析笔记（第6章）

第6章 栈，队列，优先队列 2

6-1 栈的基础、经典算法问题使用 2

 例题1：LeetCode 上第 20 题 2

 练习1：LeetCode 第 150 题 2

 练习2：LeetCode 第 71 题 2

6-2 栈和递归的紧密关系 2

 练习1：LeetCode 第144 题 2

 练习2：LeetCode 第94 题 4

6-3 使用自己编写的模拟系统栈，写出非递归的程序 4

 例题1：LeetCode 第 341 题 4

 练习1：LeetCode 第 94 题 4

 练习2：LeetCode 第 145 题 4

6-4 队列 Queue 4

 例题1：LeetCode 第 102 题 4

 练习1：LeetCode 第 107 题 5

 练习2：LeetCode 第 103 题 5

 练习3：LeetCode 第 199 题 5

6-5 BFS 和 图的最短路径 Perfect Squares 5

 例题1：LeetCode 第 279 题 5

 练习1：LeetCode 第 127 题 7

 练习2：LeetCode 第 126 题 8

6-6 优先队列 8

6-7 优先队列相关的问题（2个） 8

 例题1：LeetCode 第 347 题 8

 练习1：LeetCode 第 23 题 8

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6章 栈，队列，优先队列（18题）

栈和队列虽然是简单的数据结构，但是使用这些简单的数据结构所解决的算法问题不一定简单。在这一章里，我们将来探索，和栈与队列相关的算法问题。

6-1 栈的基础、经典算法问题使用（3题）

* 例题1：LeetCode 上第 20 题

题目要求：给定一个只包括 '('，')'，'{'，'}'，'['，']' 的字符串，判断字符串是否有效。

题目难度：简单。

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

英文网址：

求解关键：（1）括号匹配问题，使用栈。（2）使用栈弹出元素的时候，应该检查栈是否为空。

我的解答：

参考资料：

* 练习1：LeetCode 第 150 题

题目要求：逆波兰表达式求值。

题目难度：

中文网址：https://leetcode-cn.com/problems/evaluate-reverse-polish-notation/description/

英文网址：

求解关键：

我的解答：

参考资料：

* 练习2：LeetCode 第 71 题

题目要求：返回路径。

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

6-2 栈和递归的紧密关系（3题）

* 练习1：LeetCode 第144 题

题目要求：给定一个二叉树，返回它的 前序 遍历。

题目难度：

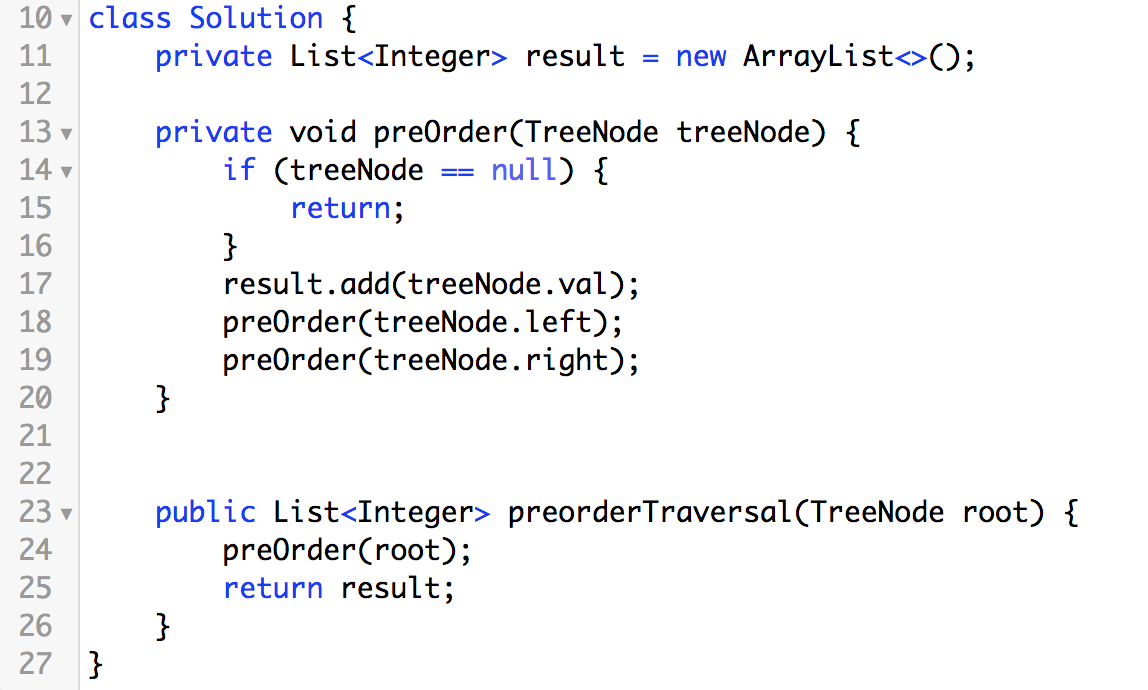
中文网址：https://leetcode-cn.com/problems/binary-tree-preorder-traversal/description/

英文网址：

求解关键：

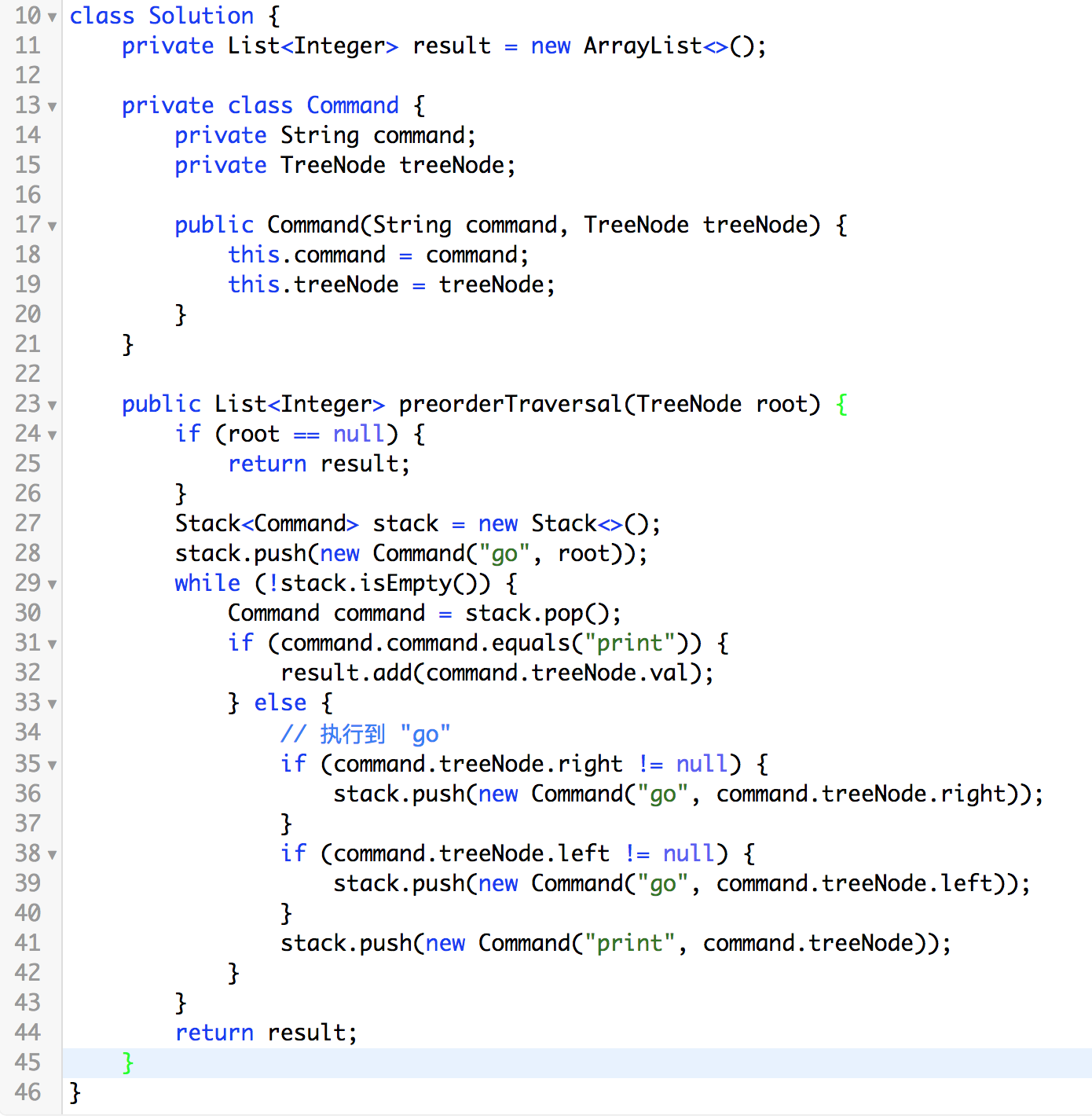
我的解答：

（1）教科书一样的递归写法：

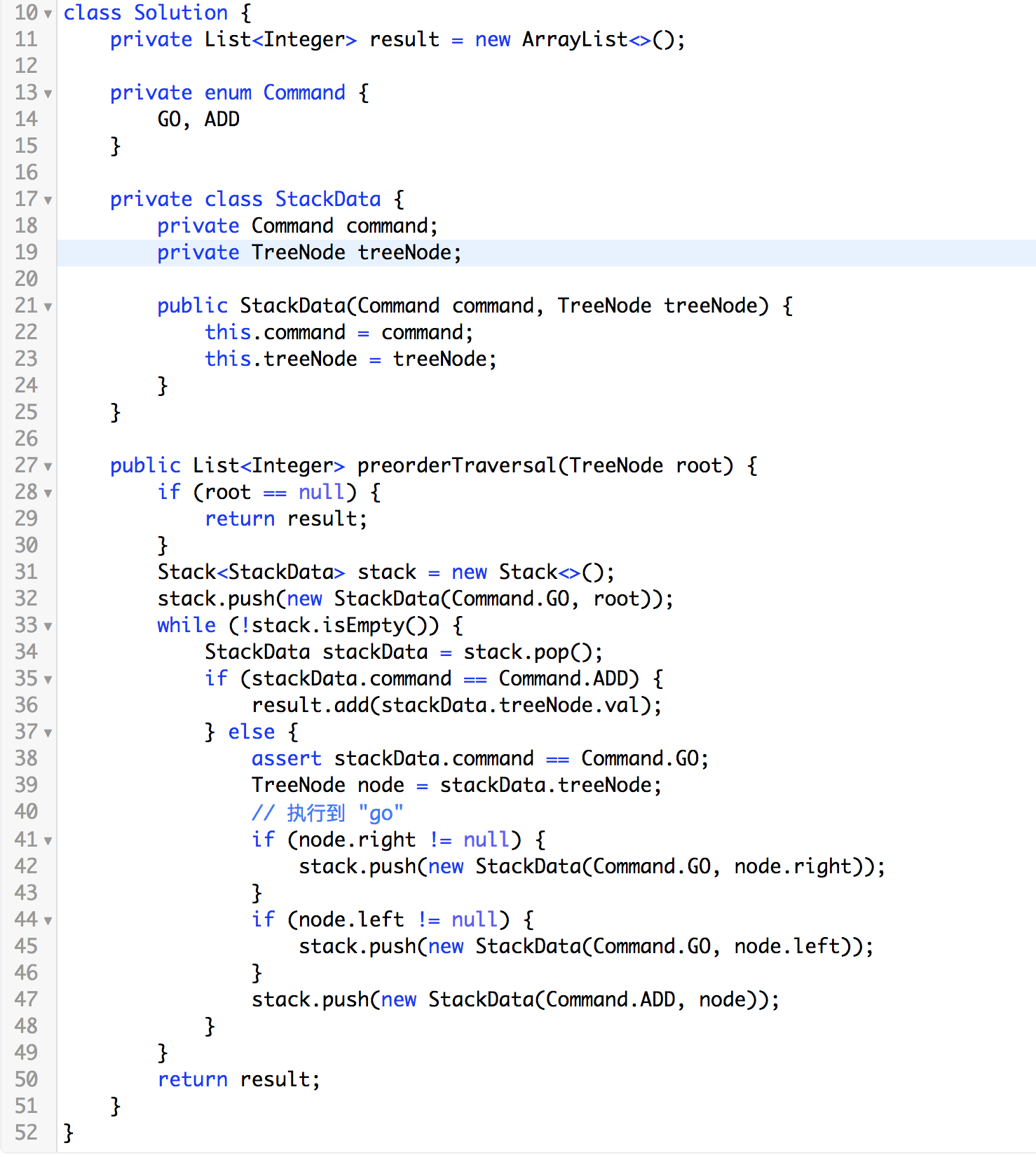


（2）模拟了系统栈的非递归写法：

马马虎虎写成了这个样子：



改了改，用枚举类：



参考资料：

* 练习2：LeetCode 第94 题

题目要求：给定一个二叉树，返回它的中序 遍历。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/binary-tree-inorder-traversal/description/

英文网址：

求解关键：

我的解答：

参考资料：

* 练习3：LeetCode 第145 题

题目要求：二叉树的后序遍历，使用递归和非递归实现。

题目难度：

中文网址：https://leetcode-cn.com/problems/binary-tree-postorder-traversal/description/

英文网址：

求解关键：

我的解答：

参考资料：

6-3 使用自己编写的模拟系统栈，写出非递归的程序（3题）

* 例题1：LeetCode 第 341 题

提示：很像系统设计

（待添加）：教科书一样的写法、邓俊辉老师的写法

* 练习1：LeetCode 第 94 题
* 练习2：LeetCode 第 145 题

6-4 队列 Queue（4题）

* 例题1：LeetCode 第 102 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：

* 练习1：LeetCode 第 107 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：

* 练习2：LeetCode 第 103 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：

* 练习3：LeetCode 第 199 题

题目要求：二叉树从右边看，得到的一个数组。

题目难度：

中文网址：

英文网址：

求解关键：

我的解答：

参考资料：

6-5 BFS 和 图的最短路径 Perfect Squares（3题）

* 例题1：LeetCode 第 279 题（油管缺少下载）

题目要求：求最小的完全平方数之和。

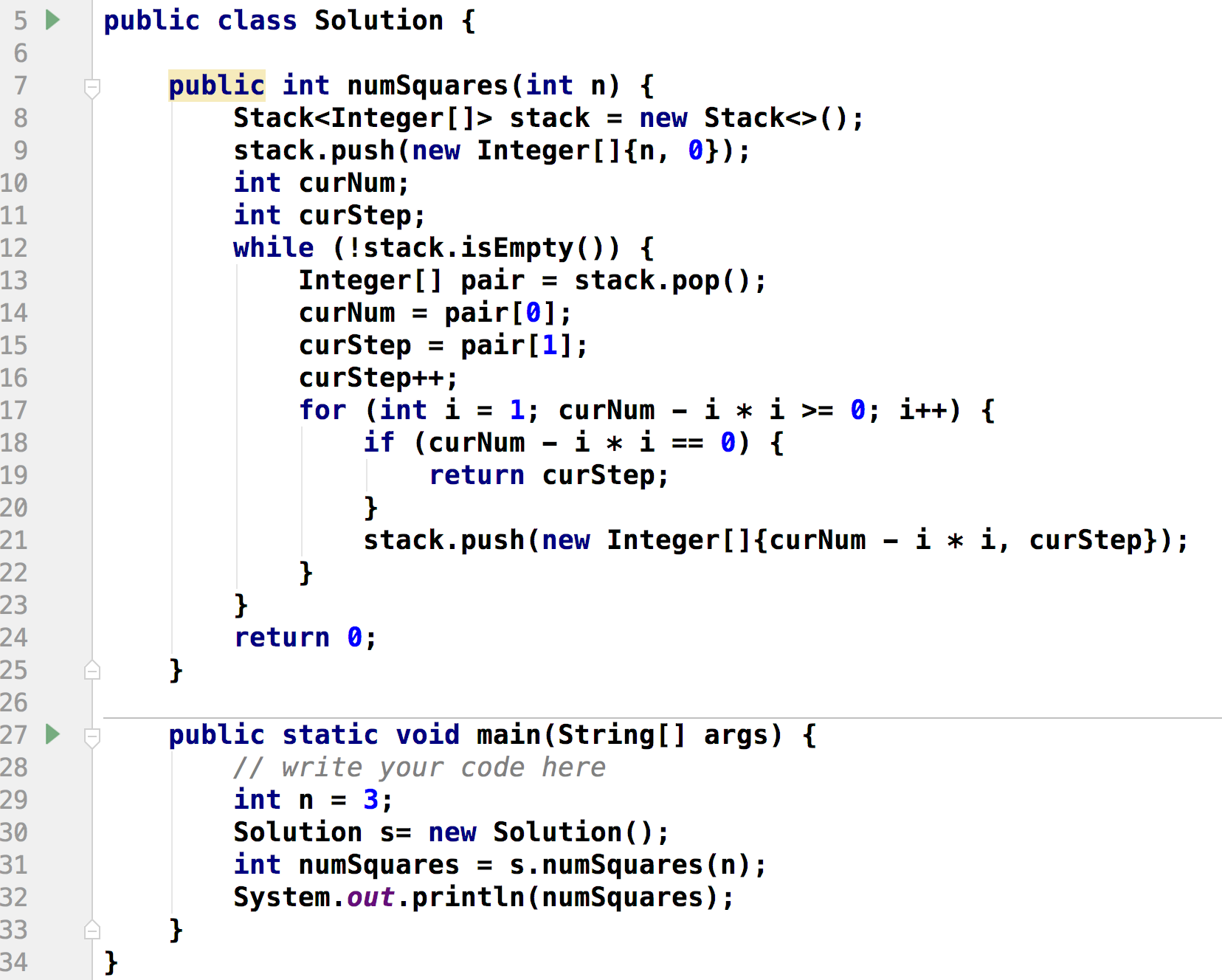
题目难度：

中文网址：

英文网址：

求解关键：建模，使用广度优先遍历。

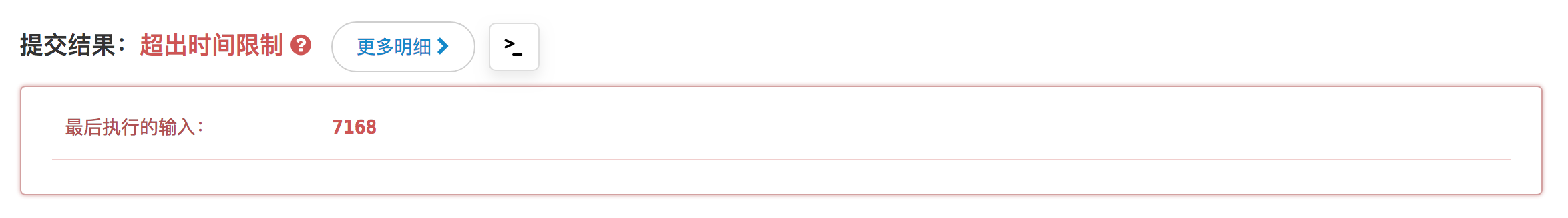
我的解答：



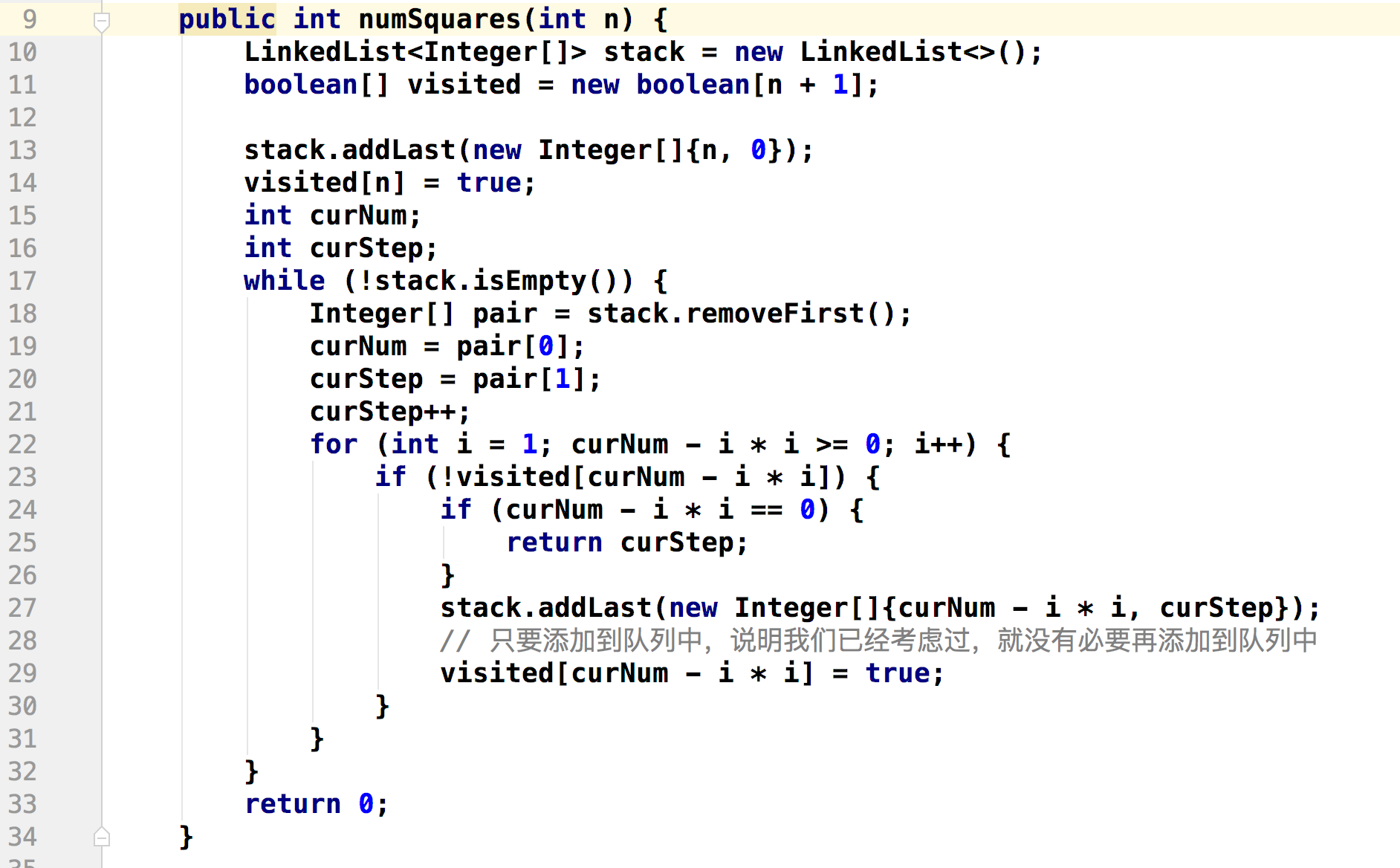
写成这样是有问题的：

（1）首先，最大的问题是，广度优先遍历应该使用队列，而不是栈（真是写代码都不过脑子！）；

（2）修改成队列以后，提交到 LeetCode 以后，发现

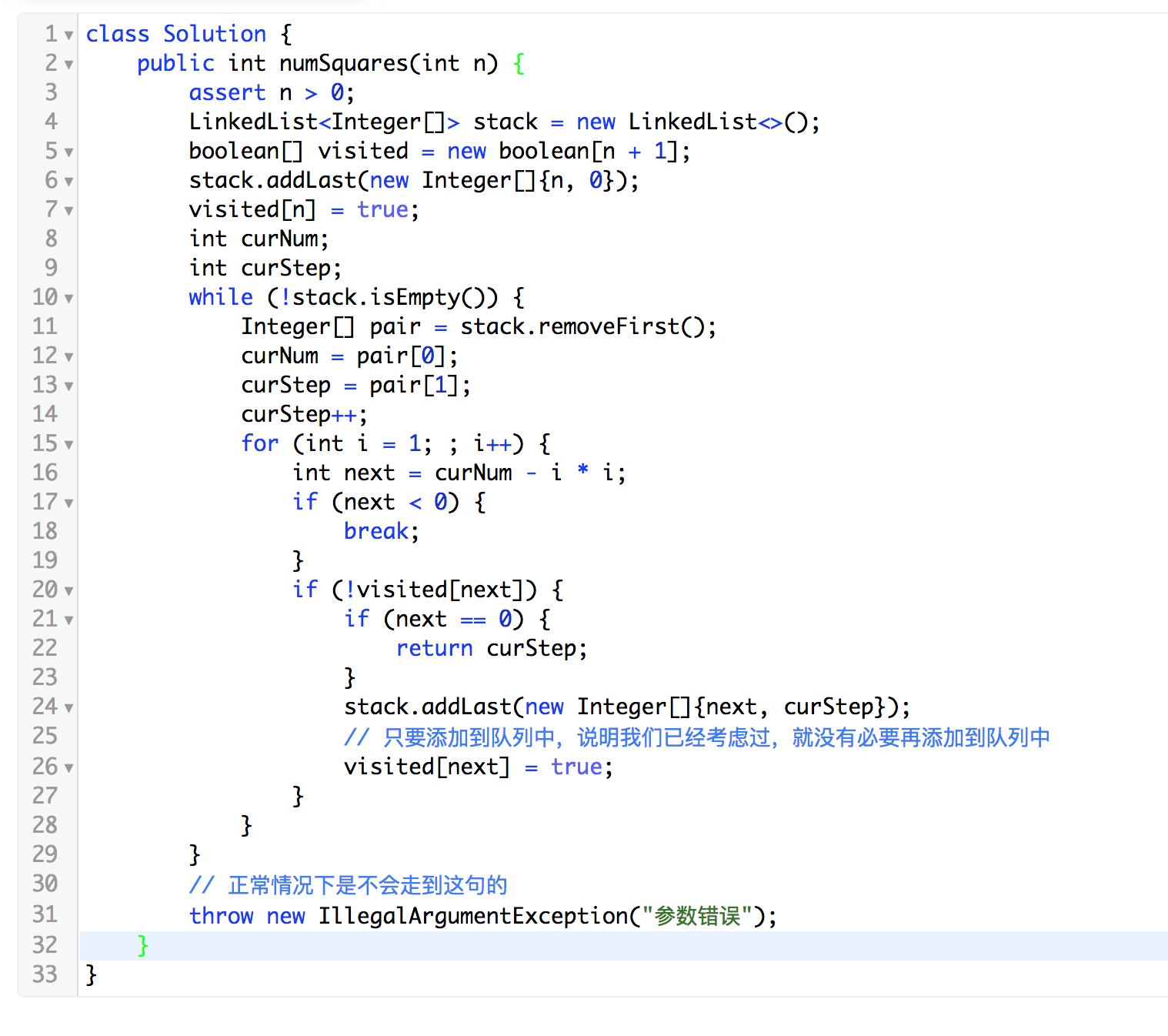


说明我们的结果是正确的，但是可能走了很多弯路，想想我们在图论中的广度优先遍历，通常都设置了 visited[] 数组，这道题也不例外。



curNum -i\*i 计算多次，应该设置局部变量。

最后，不要忘记一些异常检测。



参考资料：

* 练习1：LeetCode 第 127 题

题目要求：给定两个单词（beginWord 和 endWord）和一个字典，找到从 beginWord 到 endWord 的最短转换序列的长度。转换需遵循如下规则：每次转换只能改变一个字母；

转换过程中的中间单词必须是字典中的单词。

题目难度：中等。

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

英文网址：

求解关键：这道题目可以建模成一个有向图，原问题就可以抽象成一个有向无权图的最短路径问题，可以称之为套路的解决方案就是“广度优先遍历（BFS）”，明白了这些以后，写出正确的代码就不是难事。

注意：（1）“图”的广度优先遍历使用的辅助数据结构有两个，一个是队列（用于一层一层排队），一个是一个数组或者是集合（用于判断当前考虑的元素是否以前已经处理过）。

回顾：如果是树结构的广度优先遍历，就无需设置辅助的数组或者集合用于判断以前是否处理过。

我的解答：https://gist.github.com/liweiwei1419/8e7e0d3914e103c1d3ba4634033c9c9c

参考资料：

思考总结：其实，只要思路很清楚，写出 AC 解就是自然而然的事情了。这里我说的自然而然是相比较于要考虑很多特殊情况（边界情况，+1或者 -1 情况）的简单问题来说，这道问题应该划分到简单。毕竟广度优先遍历是一个常规问题，不需要使用到什么小技巧。

下面谈一谈我在解题过程中的一些优化：

（1）原来是遍历 wordList，看看目标 word 与它们只差一个字母的情况，此时在 wordList 很大的情况下效率低下，故将 wordList 改成 HashSet，利用 HashSet 查询高效性避免了去遍历这件事情；

（2）char[] 数组这件事情，其实还可以用 StringBuilder。（3）考虑能不能使用 Trie。

* 练习2：LeetCode 第 126 题

提示：与 127 的问题一样，保存最短路径是什么，**其实是一个标准问题**。只能使用在无权图中。

题目要求：据说是 LeetCode 上难度最大的一题，也是面试中经常考到的一题。

题目难度：

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

英文网址：

求解关键：

我的解答：

参考资料：https://blog.csdn.net/mebiuw/article/details/53165141

思考总结：很遗憾，我自己只写出了超时解。不过思路至少还是对的。先使用广度优先搜索得到最短层数，再使用深度优先搜索得到所有的最短路径。

6-6 优先队列

6-7 优先队列相关的问题（2题）

* 例题1：LeetCode 第 347 题

题目要求：前K个高频元素。注意：你可以假设给定的 k 总是合理的，1 ≤ k ≤ 数组中不相同的元素的个数。**你的算法的时间复杂度必须优于 O(n log n) , n 是数组的大小**。

题目难度：中等。

中文网址：https://leetcode-cn.com/problems/top-k-frequent-elements/description/

英文网址：

求解关键：“**你的算法的时间复杂度必须优于 O(n log n) , n 是数组的大小**”。这是题目对我们的要求，我们很容易想到的一种思路是将 counter 以后的数据对 value 进行排序，但即使是最好的排序算法，时间复杂度也是 **O(n log n)** ，换言之，题目限制了我们不能使用排序算法。那么，对于前 k 这样的问题，一个很自然的思路就是使用优先队列，想到这一点，这道问题就是一个常规问题了。

我的解答：

参考资料：

思考总结：

* 练习1：LeetCode 第 23 题

题目要求：

题目难度：

中文网址：https://leetcode-cn.com/problems/merge-k-sorted-lists/description/

英文网址：

求解关键：这里我们举生活中的例子，其实一点都不难。

假设有如下生活情境：假设你是一名体育老师，**假设三个班的学生已经按照身高从矮到高排好了**，现在要对 3 个班的学生的身高进行排序。

（1）让三个班的学生按列站在你的面前，你能看到队首的学生的全身，其余同学只能看到比前面同学脑袋高出的那部分；

（2）每一次队首的 3 名同学，请出最矮的同学出列到“队伍4”（即我们最终认为排好序的队列），出列的这一列的后一名同学向前走一步；

（3）重复第（2）步，直到 3 个班的同学全部出列完毕。

我的解答：

参考资料：

思考总结：