1. Two Sum

10. Regular expression matching

13. Roman to Integer

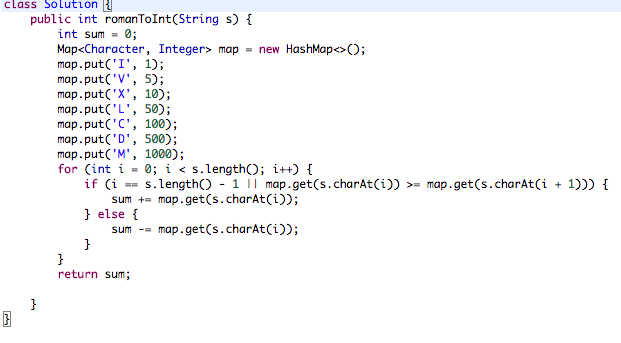
题意：把罗马数字变成数字 时间复杂度 o(n) 空间复杂度，罗马字数对应得个数

解题：

1.先把 每个罗马字对应得数字放入('I', 1)， （'V', 5) ('X', 10) ('L', 50)（'C', 100)('D', 500) ('M', 1000)-----Map也可以用一个长度为26的array来代替

2. 如果一个小的数字出现在了大的数字的左边，那么减去这个数，不然加上这个数

比如CM（900），C 先出现减去100，然后M 出现加上一千，就是900



20. Valid Parentheses

时间o(n),空间stack o(n)

1.a[i]是右括号

把对应的左括号放进去

1. 是左括号，判断stack不为空，且pop出来的值和左括号相等

3.最后判断stack是否为空

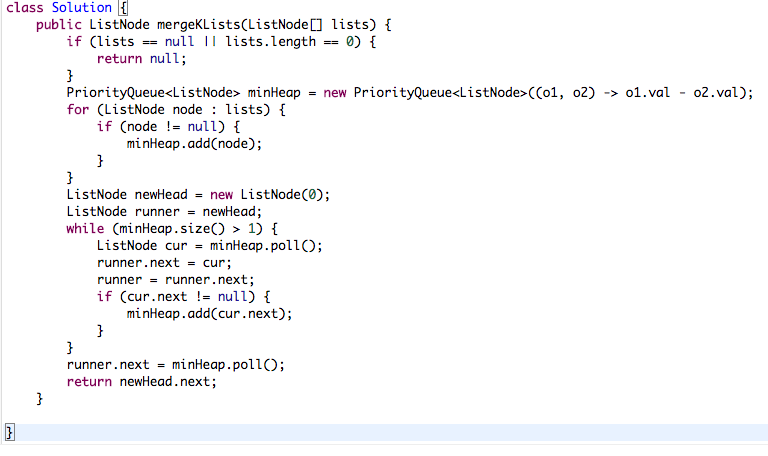
**#23. Merge k Sorted Lists**

题意: Given k 个sort好的list，合并成一个list

解题1：每次找到头顶最小的元素，加入新的list中，如果这个list下一位不为空，放回原来的poll里继续比较，为了快速找到最小的元素使用MinHeap。

时间复杂度：假设总共有k个list，每个list的最大长度是n，这样每次找到最小元素的时间为logk，时间复杂度为nklogk

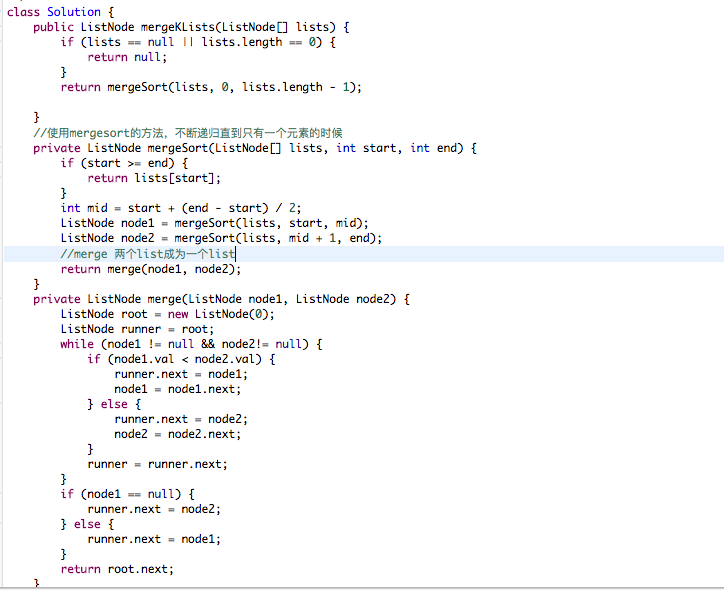
空间复杂度：栈的大小o(k)



解题2：用merge sort

时间复杂度： 那么运行时间满足递推式T(k) = 2T(k/2)+O(n\*k)。根据主定理，可以算出算法的总复杂度是O(nklogk)。

空间复杂度：递归栈的大小O(logk)??



91Decode Ways

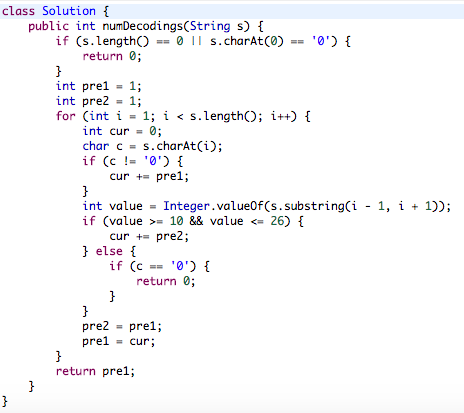
Method1: 以a[i]为结尾的可能性

如果s[0] 为0，直接返回0

1 a[i] != 0,存在以结尾的一个可能 cur += pre1

2.1a[i-1]~a[i]小于等于26，大于等于10，存在以i结尾的两位数 cur += pre2

2.2 如果大于26，小于10，且为0，直接返回0



Method2: 以a[i] 为开头的可能性

1. a[i] ==‘0’continue,不可能以0开头

2. a[i] !=‘0’

2.1 如果a[i]-a[i + 1] 小于等于26，大于等于10，证明能以a[i]为开头形成一个两位数

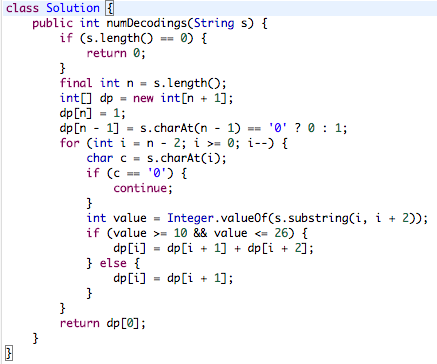
f[i] = f[i + 1] + f[i + 2]

2.2 不能形成一个两位数，只能分别成为两个一位数

f[i] = f[i + 1]

i基于后面的结果所以i-- loop

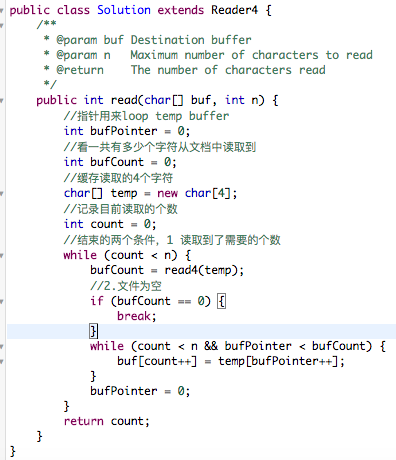
Base case: dp[n] = 1, dp[n – 1] = charAt(n-1) != ‘0’ ? 0 : 1



157. Read N Characters Given Read4

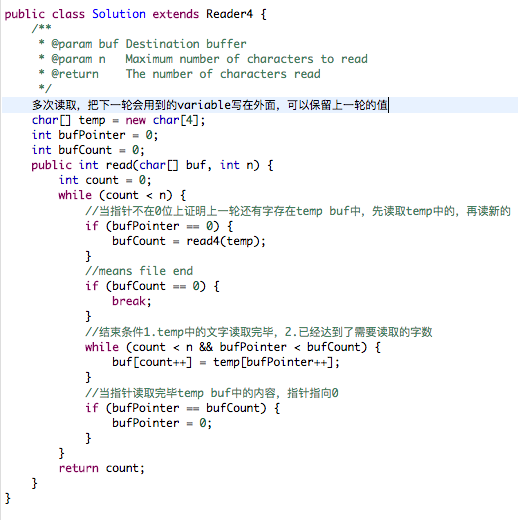
题目大意：给定一个buffer array，和需要读取字数n，和一个read function API，返回一共读取的字数

（read的function，每次可以读取4个character，参数是一个char array，传进去后读取的文字会输入在array里，且function会返回一个读取到的个数，如果file的结尾只剩下3个那么只会读取3个）



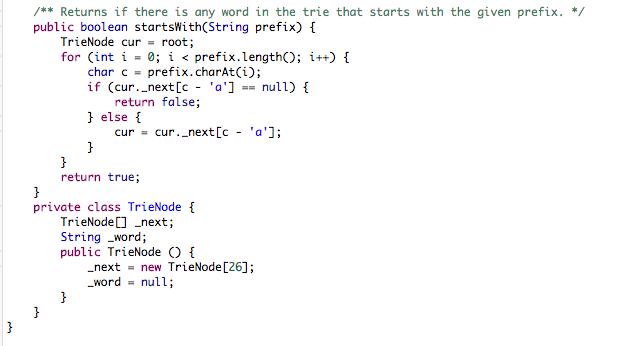
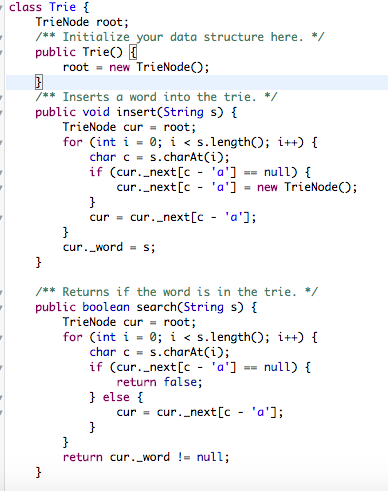
158. Read N Characters Given Read4 II

允许多次读取，需要保留上一次没有读取完全的值，



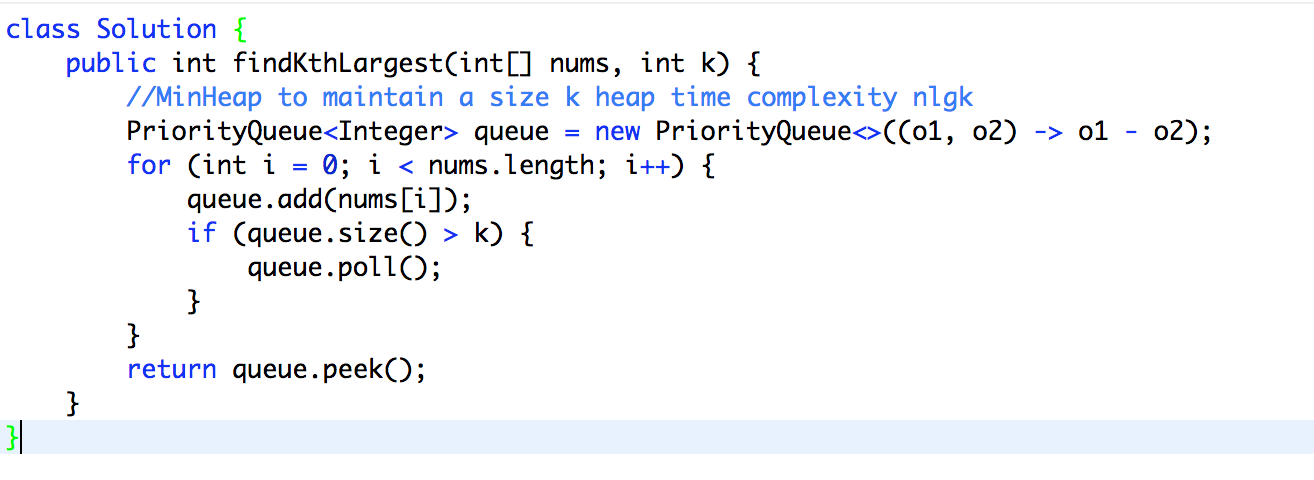
 208. Implement Trie (Prefix Tree)

题意：创建一个trie tree，实现inset， search和prefix功能



215. Kth Largest Element in an Array

Method1: priority queue time complexity nlgk space k

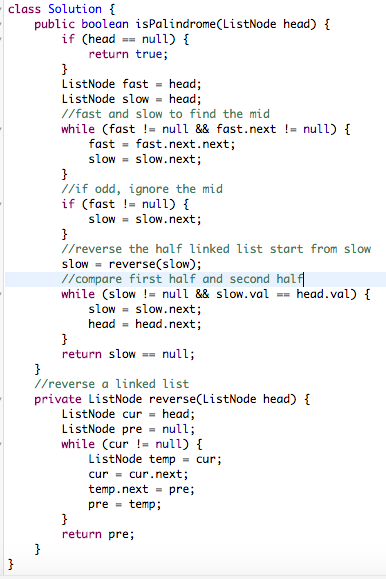


Method 2: quicksort

234. Palindrome Linked List

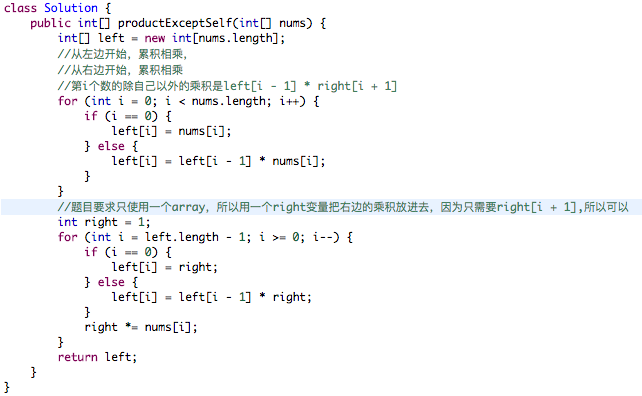
题意，Given 一个linkedlist，看里面的value是不是回文

用快慢指针找到中间，如果是偶数，忽略中间位，然后reverse后半段，从前开始依次比对



238. Product of Array Except Self

给定一个array，算出除了本身数字之外的所有数字的乘积



253. Meeting Rooms II

Given array of meeting time, find out the minimal number of room needed

先按meeting的开始时间排序

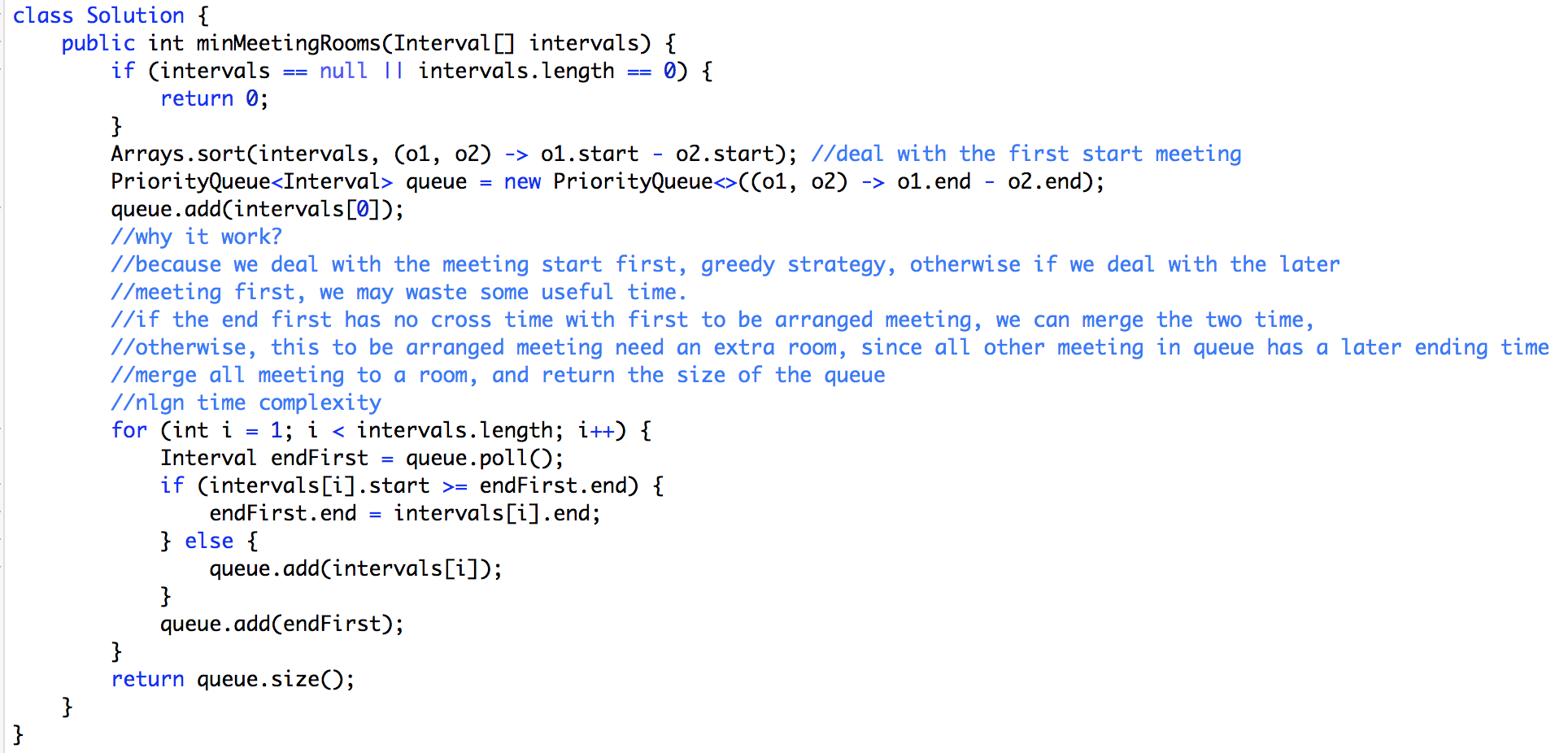
然后用一个priority queue排序已经安排好的会议的结束时间，先结束的在前

把第0个会议放进queue里.

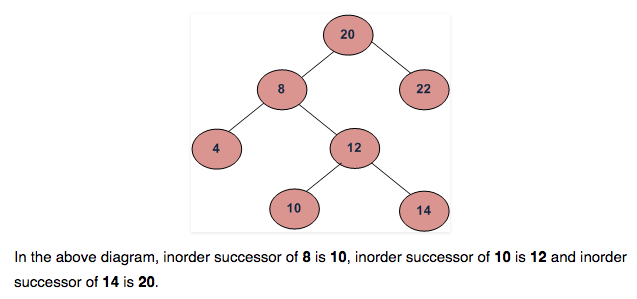
依次开始比较后面的会议，如果开始时间晚于第一个结束的时间，那就合并两个会议，放回queue里，不然需要一个新的meeting room，把当前会议和queue poll出来的会议都放回queue里供后面的会议比较

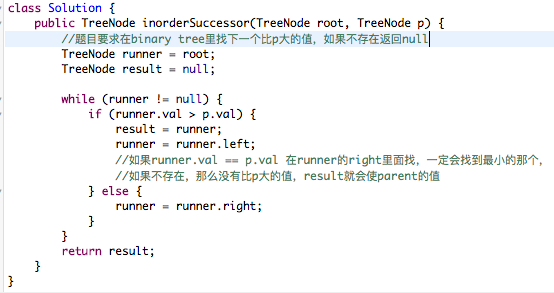
最后的结果就是queue的size

时间复杂度为 ： nlgn



285. Inorder Successor in BST





297. Serialize and Deserialize Binary Tree

题意把Tree编译成一串字符，然后按规则把字符还原成树

Method 1 : BFS + queue

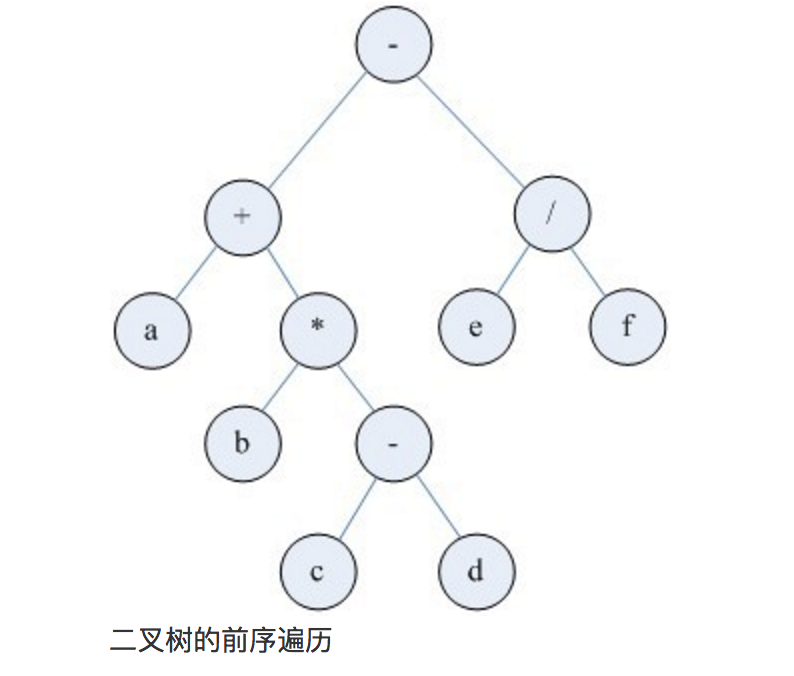
Serialize: 按前序遍历，把node按层依次加入List里，如果为叶子节点，需要加入“n “表示为叶子节点，最后倒着把list里的 “n ”去掉。注意依次加入每个点时，加入空格，方便deserialize 时split成array

Deserialize: 按层依次遍历，还原树，先把string split成一个array，分隔符为空格

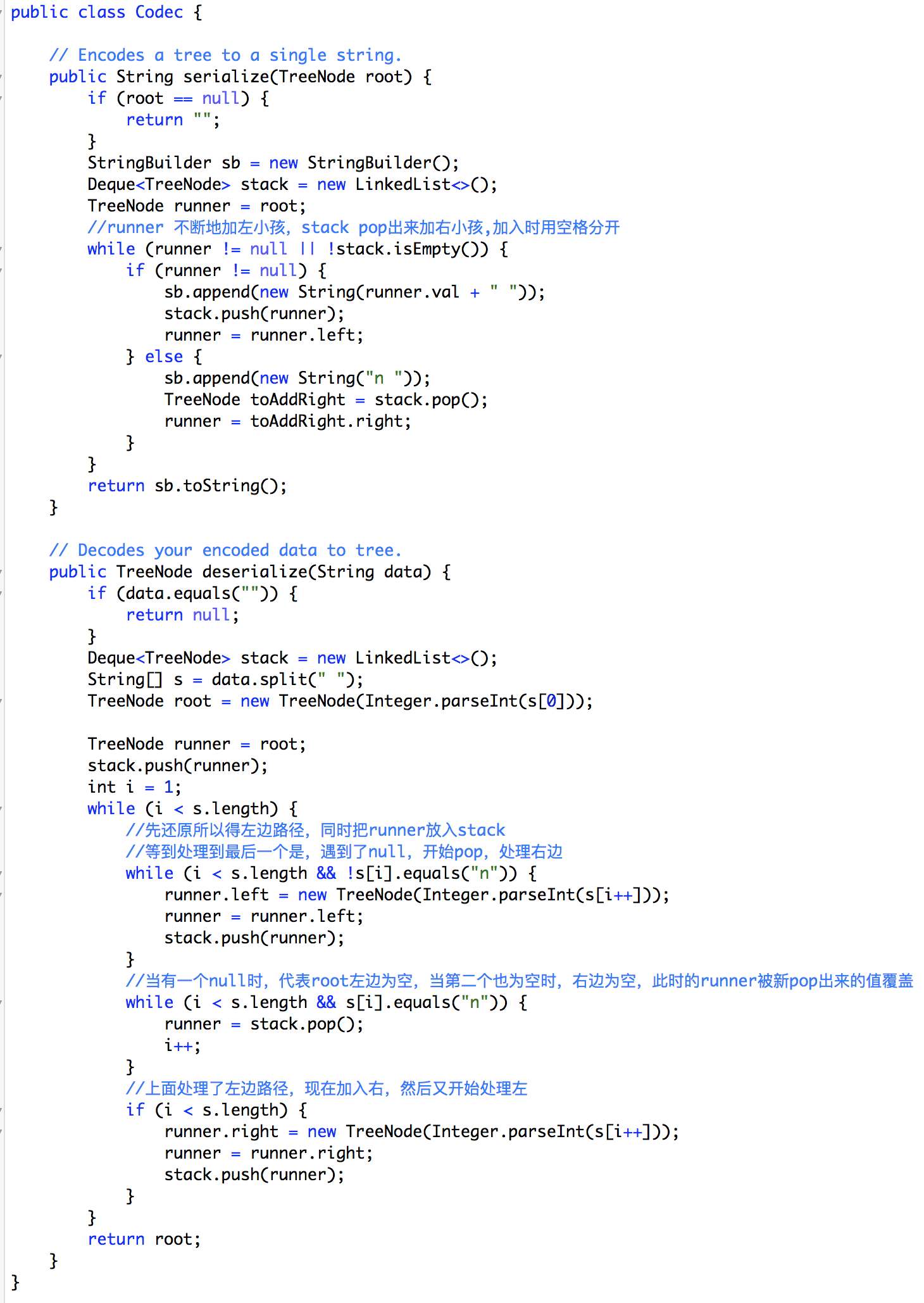
Array的第一个元素就是root，把root放入queue中，用for loop遍历剩余的元素，每次遍历从queue中poll出一个node，和array中的两个元素进行匹配，如果array中的第一个元素不为n,那么node的left就指向它，同时把这个left放入queue中，等待处理， 然后i++, 看下一个array是不是n， 如果不为空，就为node 的right， 也放入queue中



Method 2 ： preorder + DFS ，只有前序遍历能找到root

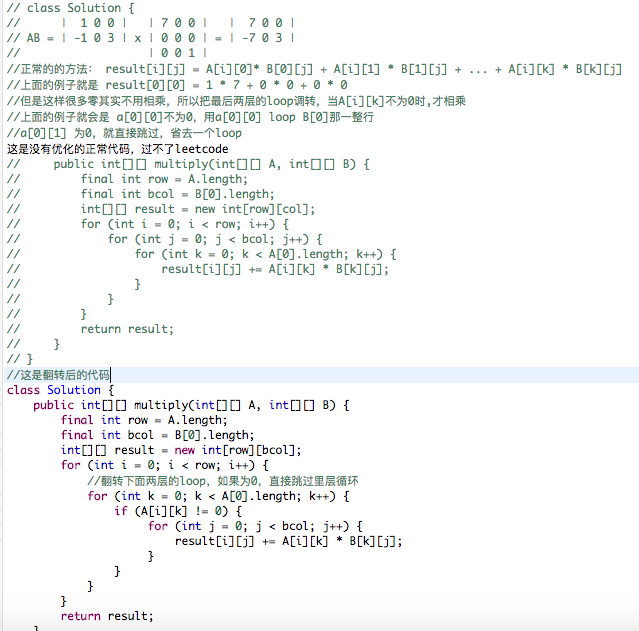


前序遍历：- + a \* b – c d / e f  
中序遍历：a + b \* c – d – e / f  
后序遍历：a b c d – \* + e f / -



311. Sparse Matrix Multiplication

题目大意：given 两个矩阵，里面有很多的0，怎么高效的得到矩阵的乘积



follow up: 一个面经的题 Sparse Vector dot Product

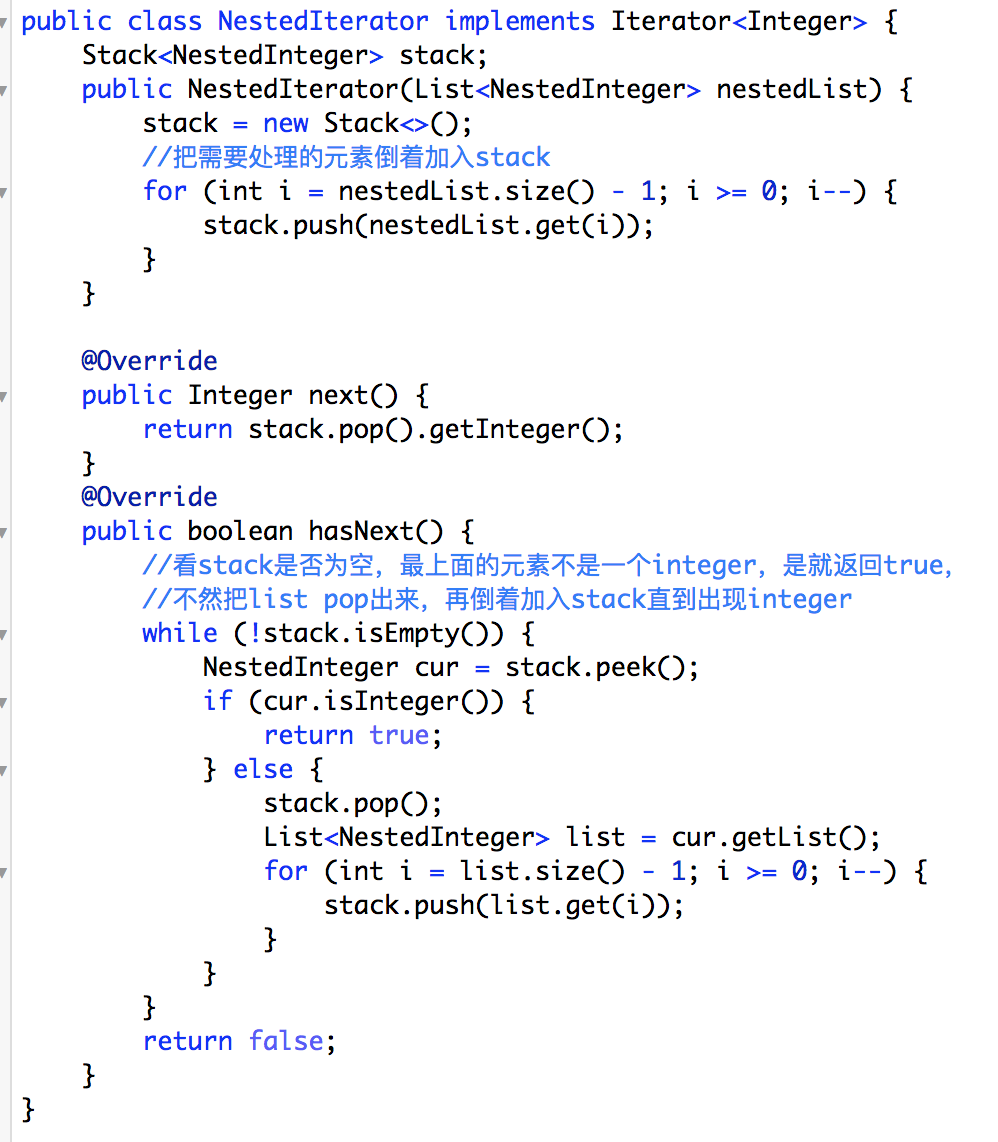
Suppose we have very large sparse vectors, which contains a lot of zeros and double.

1. find a data structure to store them
2. get the dot product of them

341. Flatten Nested List Iterator

Given nested list, 把nested list处理成一整个list

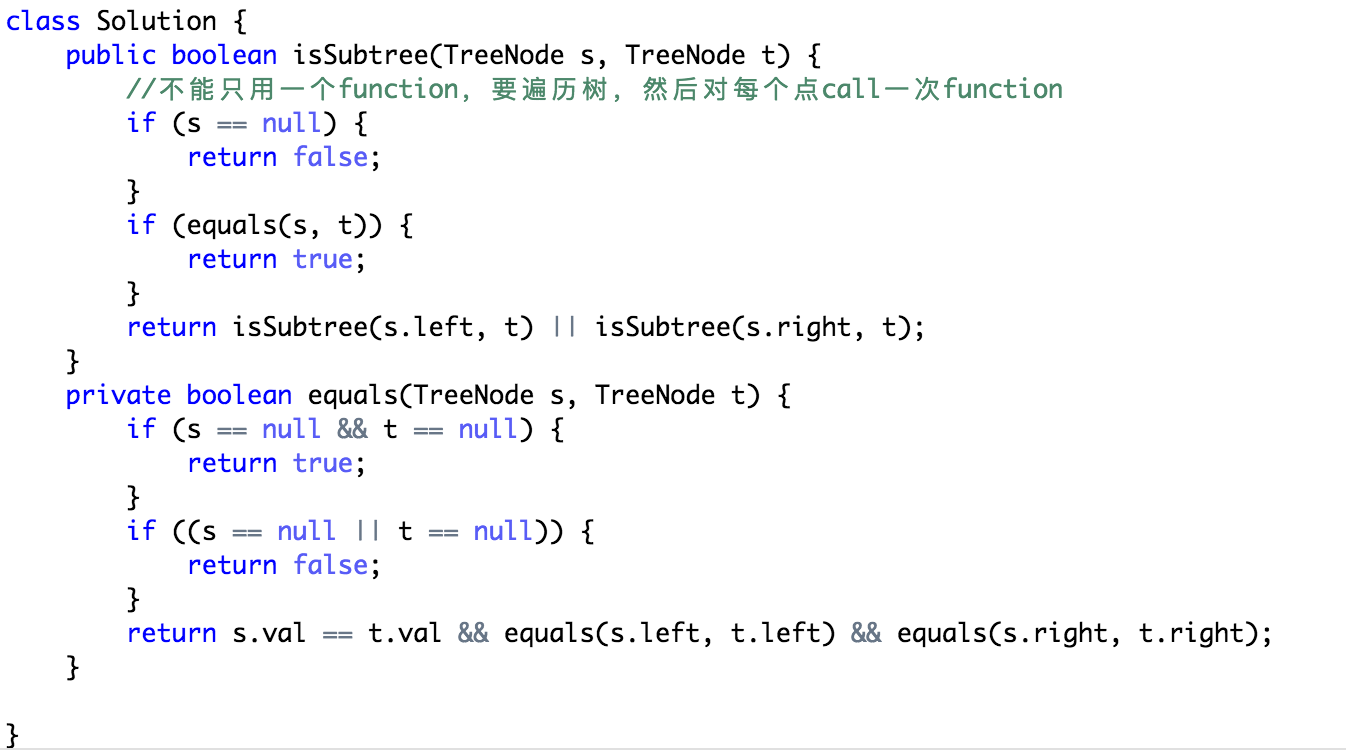
策略：不要一开始就把所有的nestedlist处理成flat，先都倒着放进stack里，等需要的时候在进行处理



572. Subtree of Another Tree

Method 1:

遍历树的每个点，对于每个点都call一次比较的function，worse case O(m \* n)



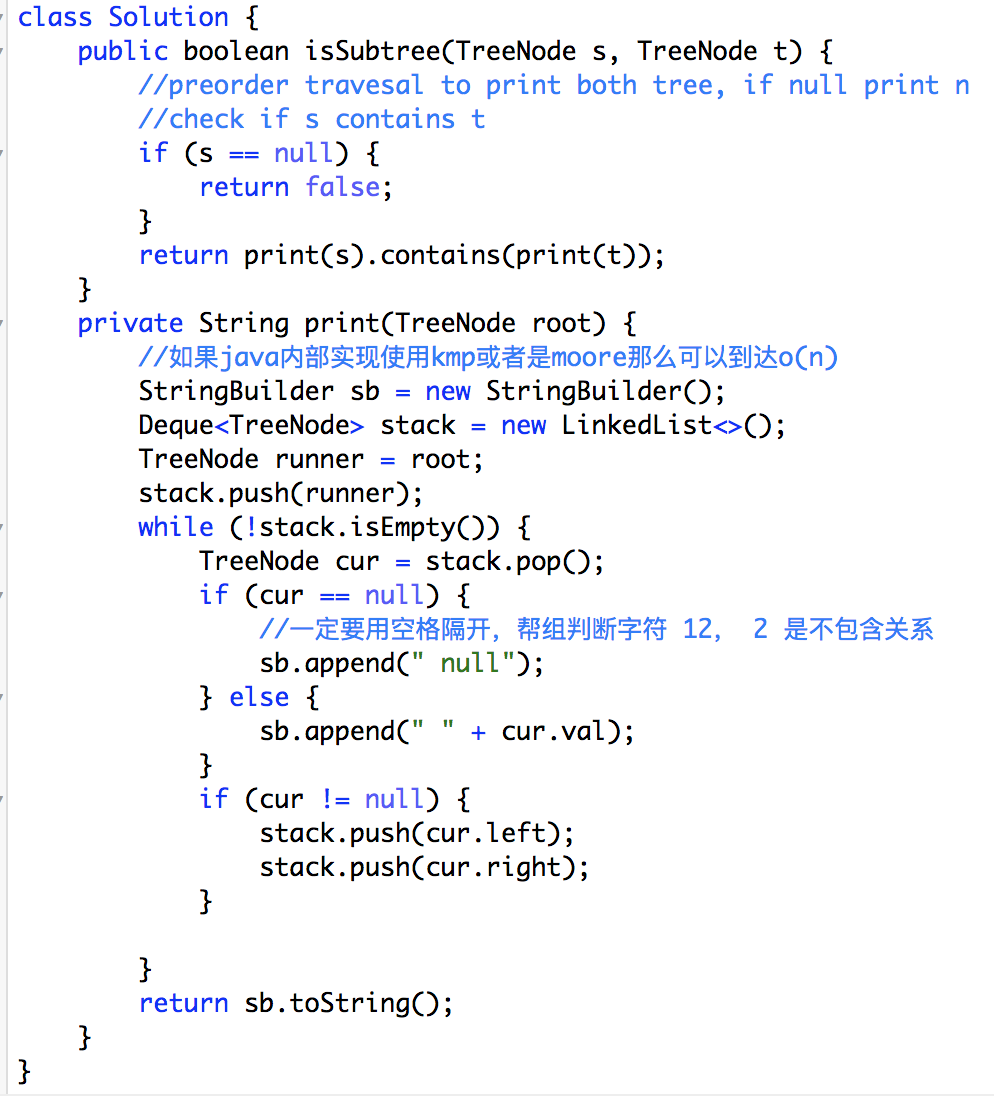
Method2:

用preorder的遍历打印两棵树，每个node的前面加上空格分开

最后看s是否包含t

复杂度： 如果使用KMP或者Boyer-Moore可以达到o(n), 暴力破解n方

Boyer-Moore technique match the characters from right to left, works well on long patterns. knuth moris pratt match the characters from left to right, works fast on short patterns.



639. Decode ways II

f(i) = ways[i] \* f( i– 1) + ways[i – 1 ~ i] \* f(i – 2);