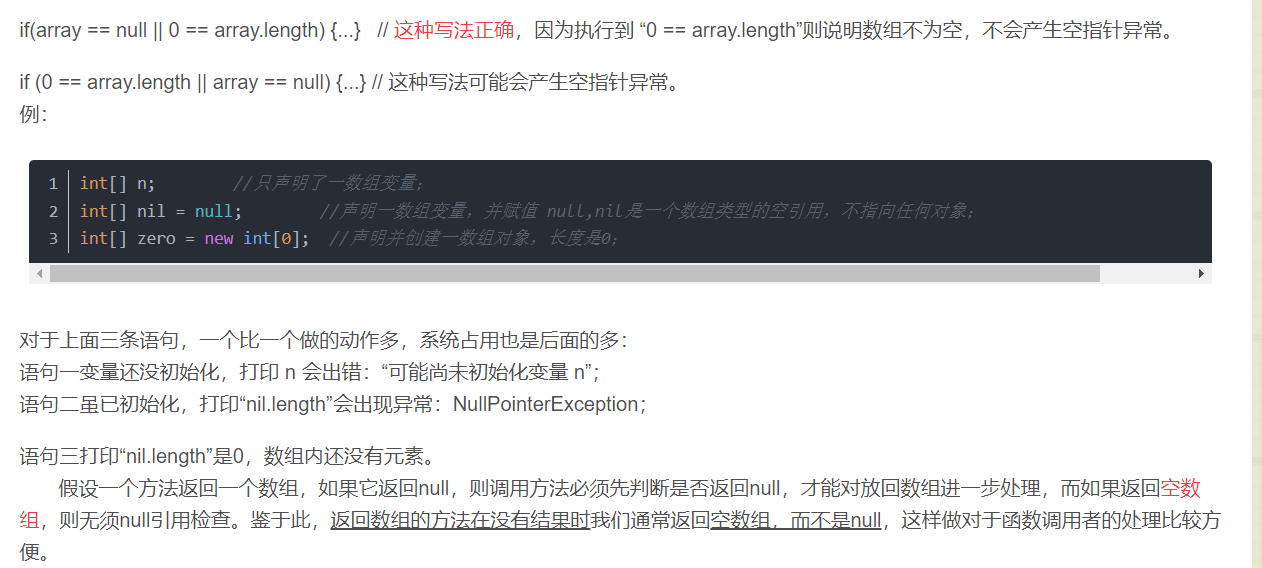
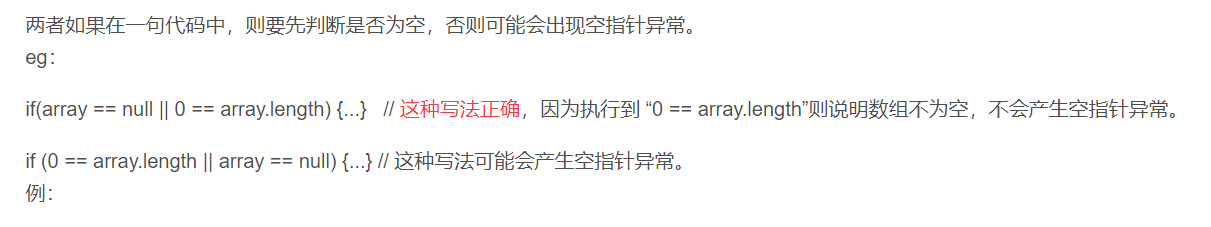
Java中栈的使用：





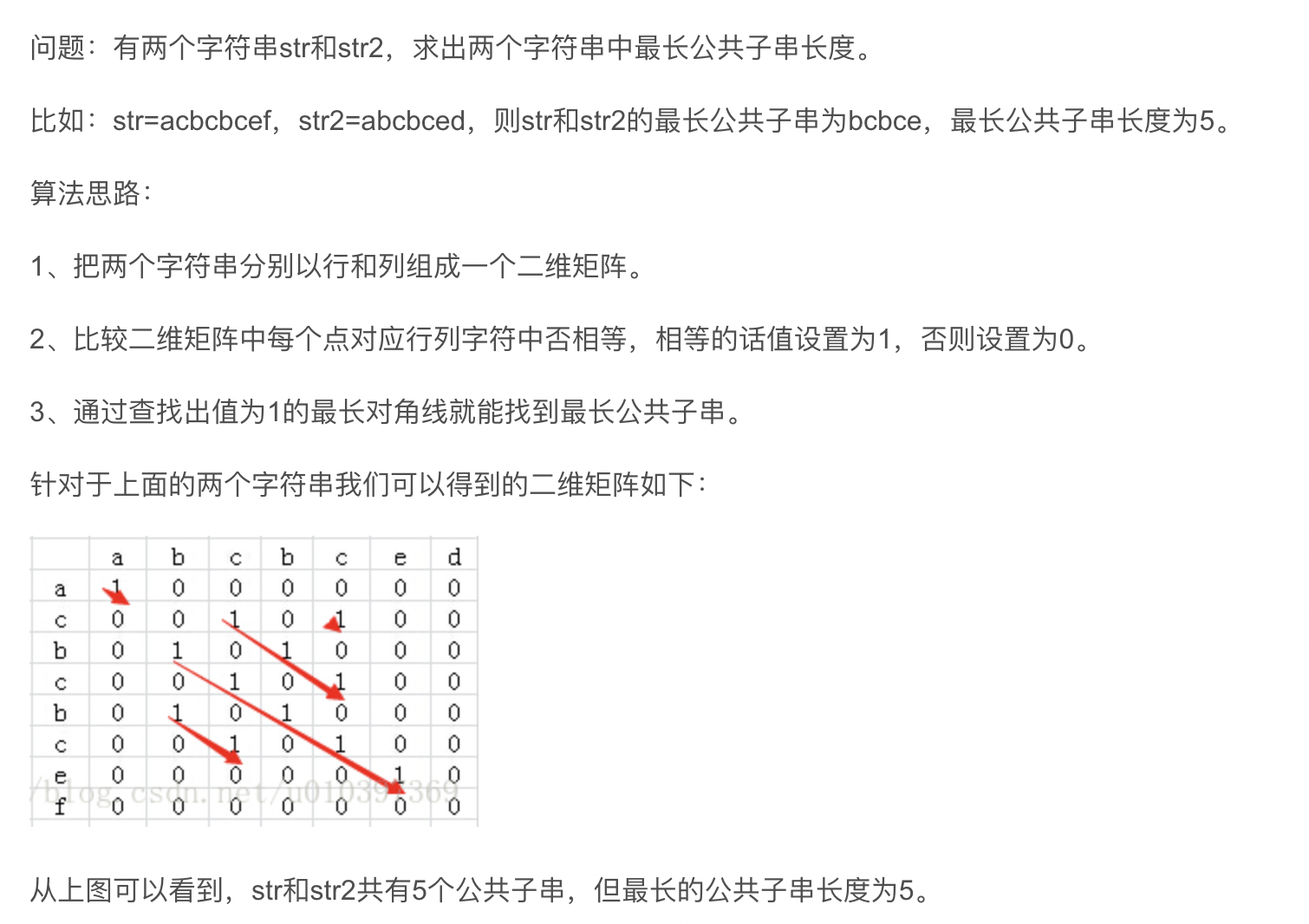


Hash map的使用，减少一层for循环，将时间复杂度降至O(n)



Leecode Q5 <https://leetcode.com/problems/longest-palindromic-substring/>

最长公共字串

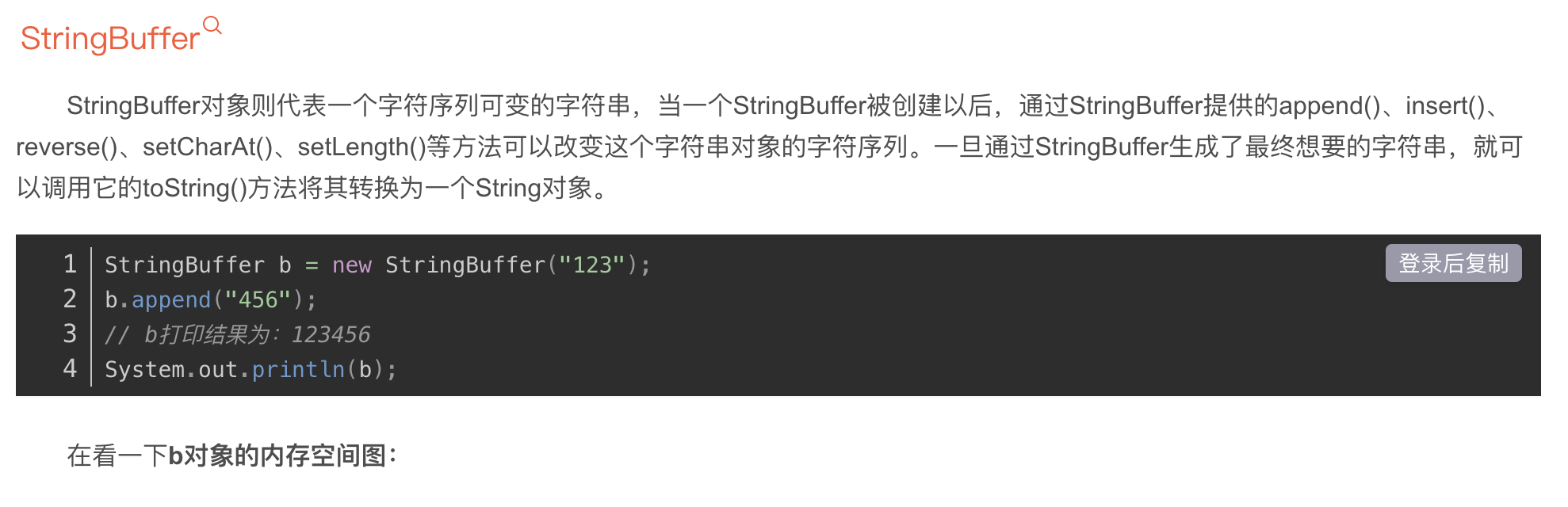


Leecode 006

List







StringBuffer类中的方法都添加了**synchronized关键字**，也就是给这个方法添加了一个锁，用来保证线程安全。

StringBuilder类也代表可变字符串对象。实际上，StringBuilder和StringBuffer基本相似，两个类的构造器和方法也基本相同。不同的是：**StringBuffer是线程安全的，而StringBuilder则没有实现线程安全功能，所以性能略高**

**Idea 替换快捷键 control+f+r**

**数组排序：**

**Arrays.sort(arr)**

**获取排序前数组的序号：**

Map<Integer,Integer> result=new HashMap<Integer,Integer>();

**Int [] arr2=arr.clone();**

**Arrays.sort(arr2);**

**Int i=0;**

**For(int data:arr2){**

**Result(data,i++);**

**}**

**Int [] index=new int[arr.length];**

**i=0;**

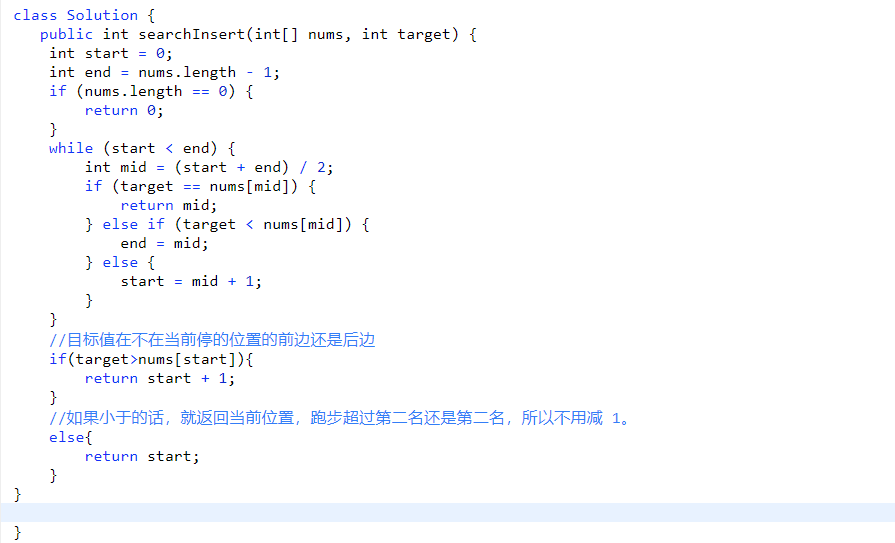
**For(int data:arr2){**

**Index[I++]=data;**

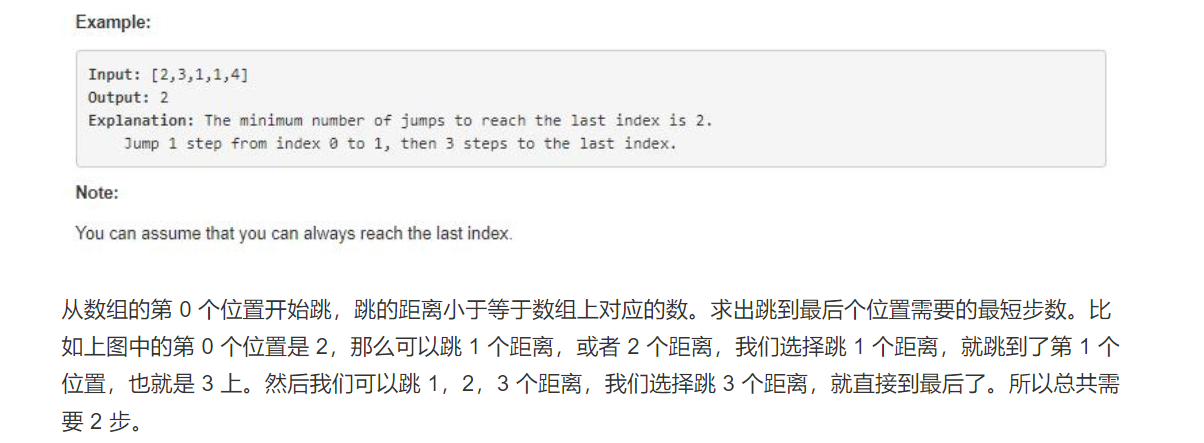
**}**

**存在缺点，无法处理有相同数字的数组。**

**二分查找理解：**



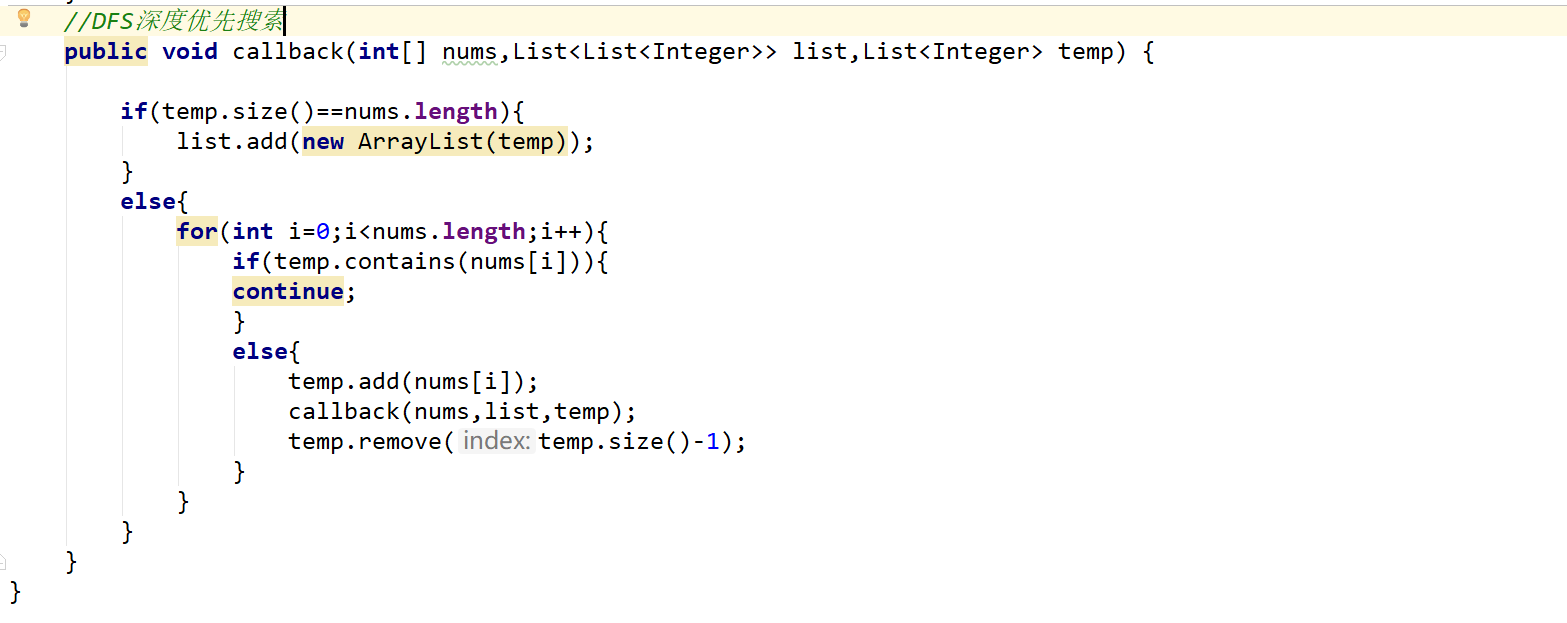
Target值最终落在start的两侧。

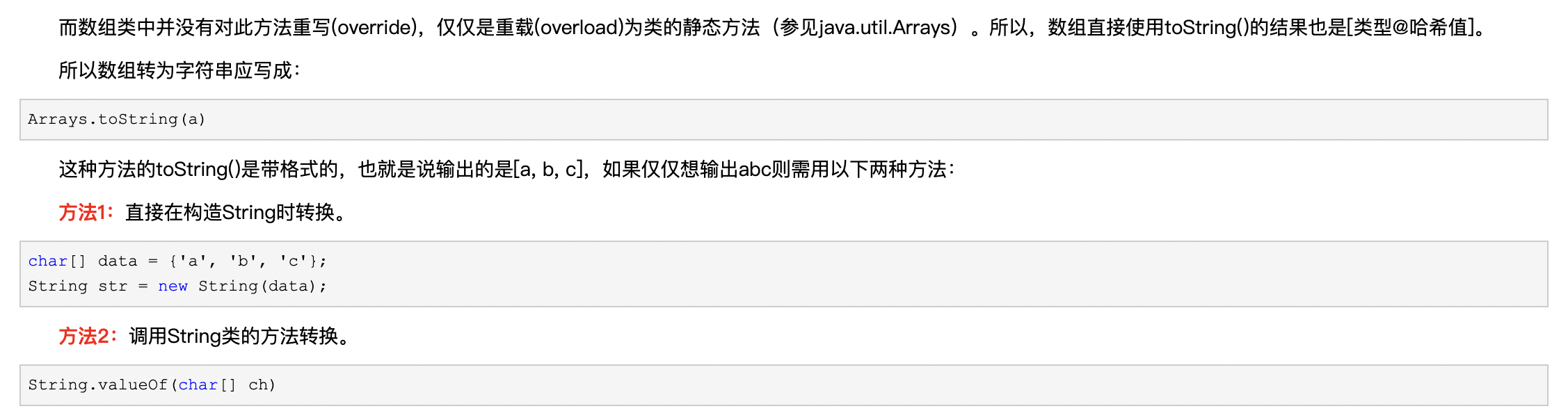


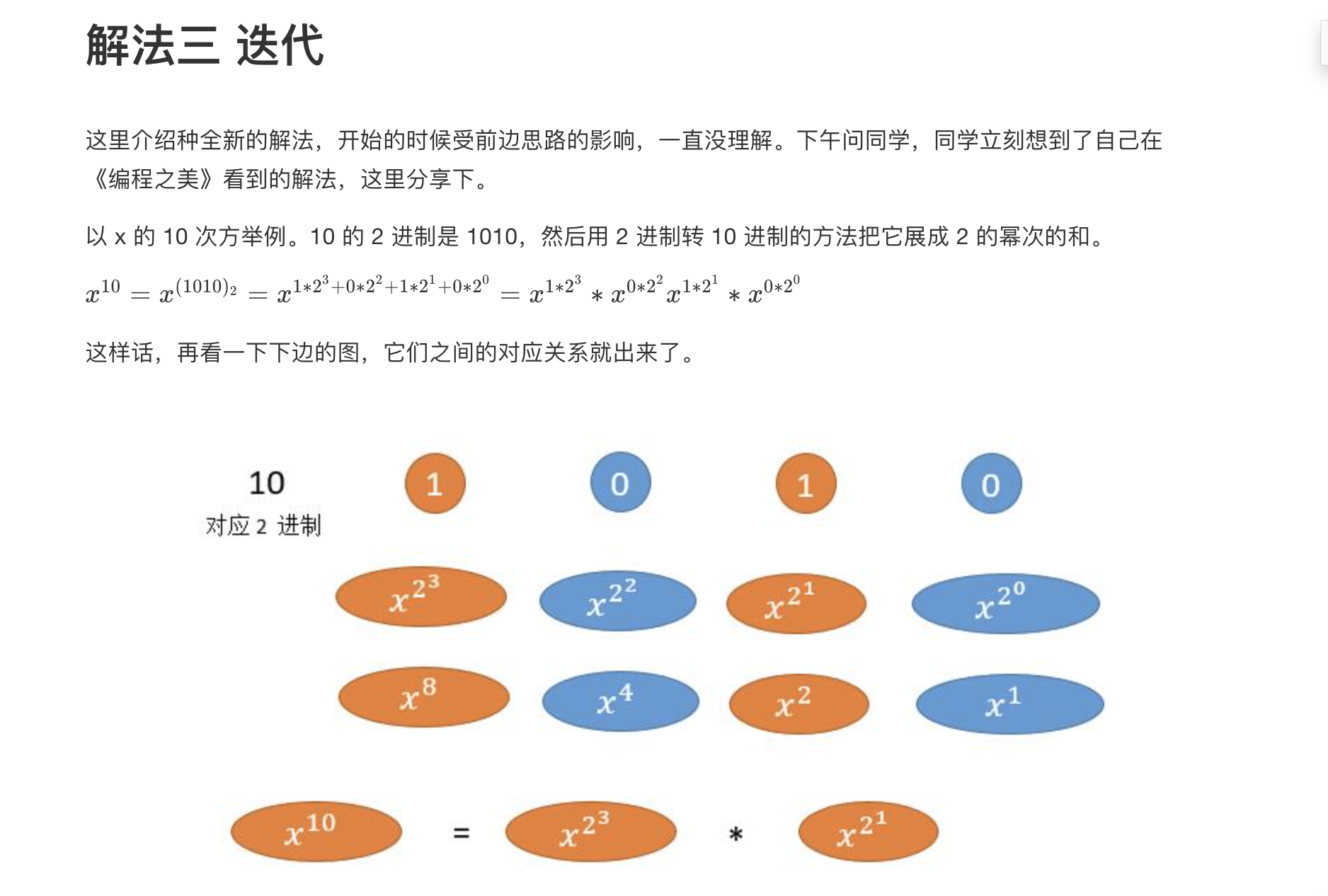


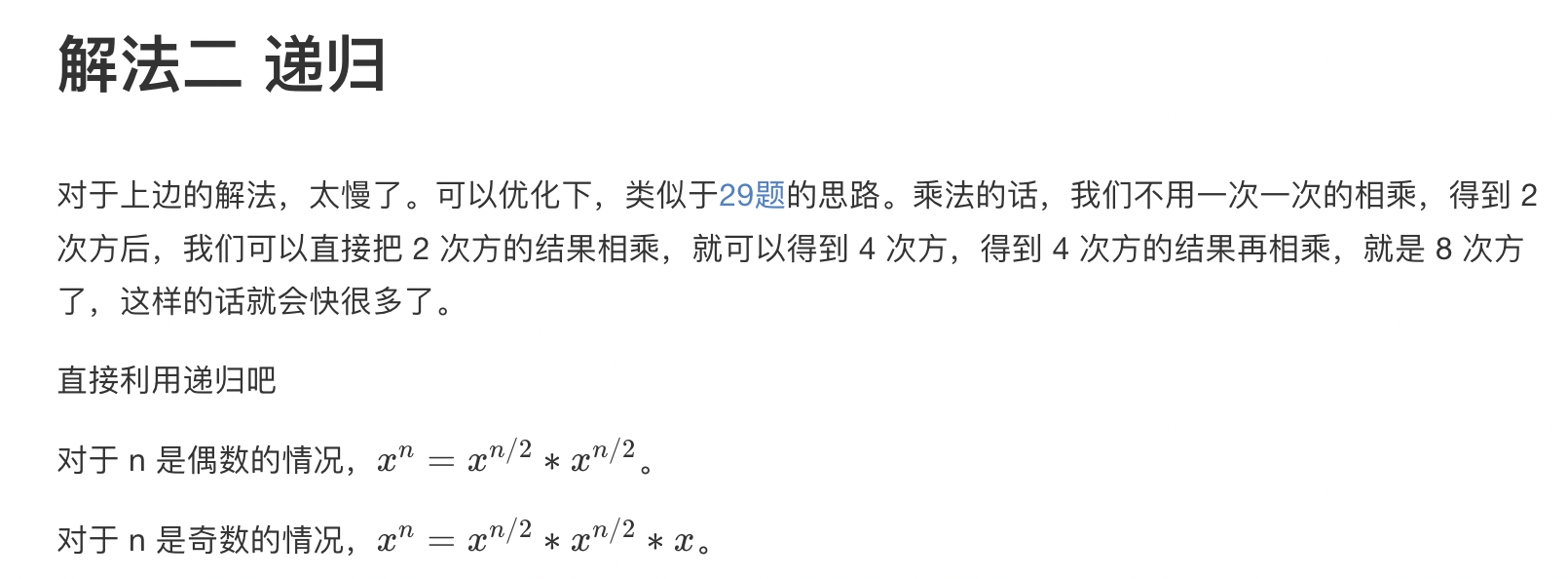


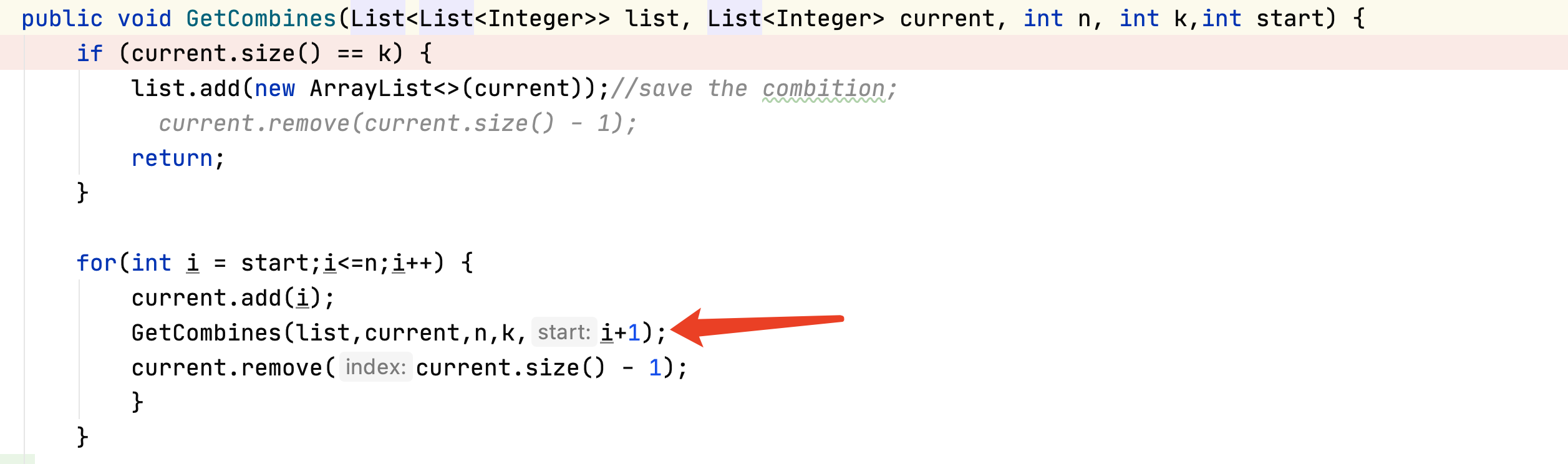




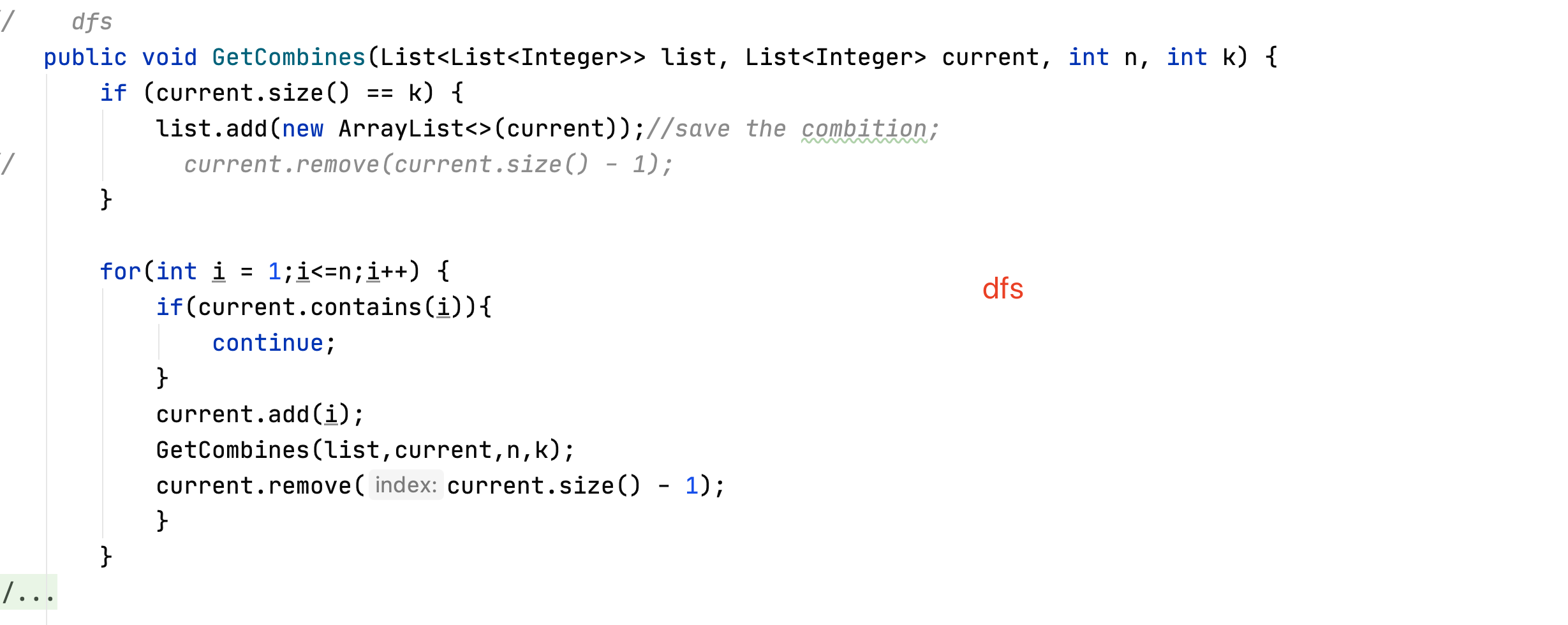








(i+1)时，返回不重复的排序；

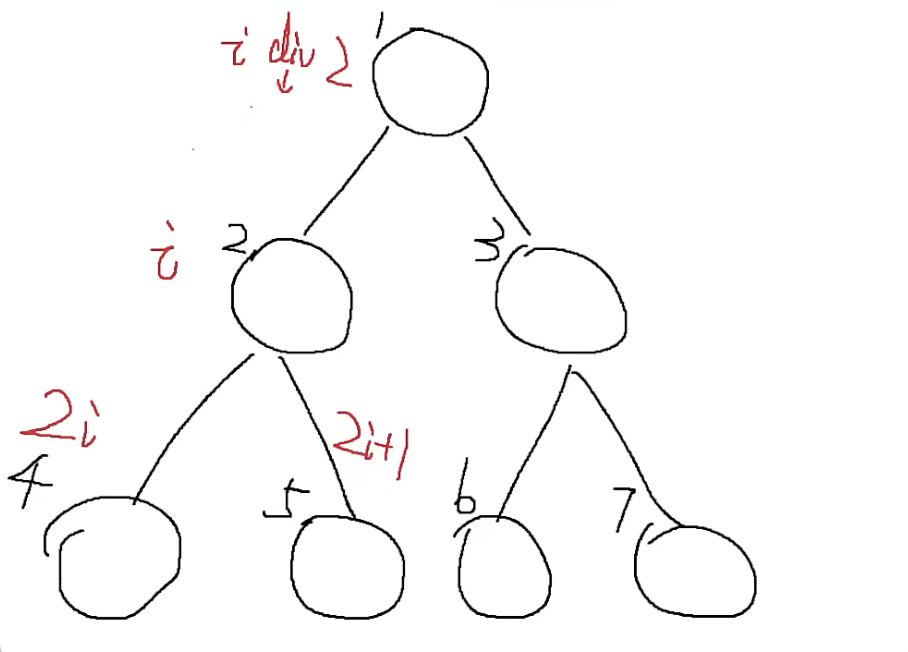




快慢指针



线段树：问题需要满足区间加法



---------------------------

技巧篇：

链表复制值时，需要记住前面节点的值。

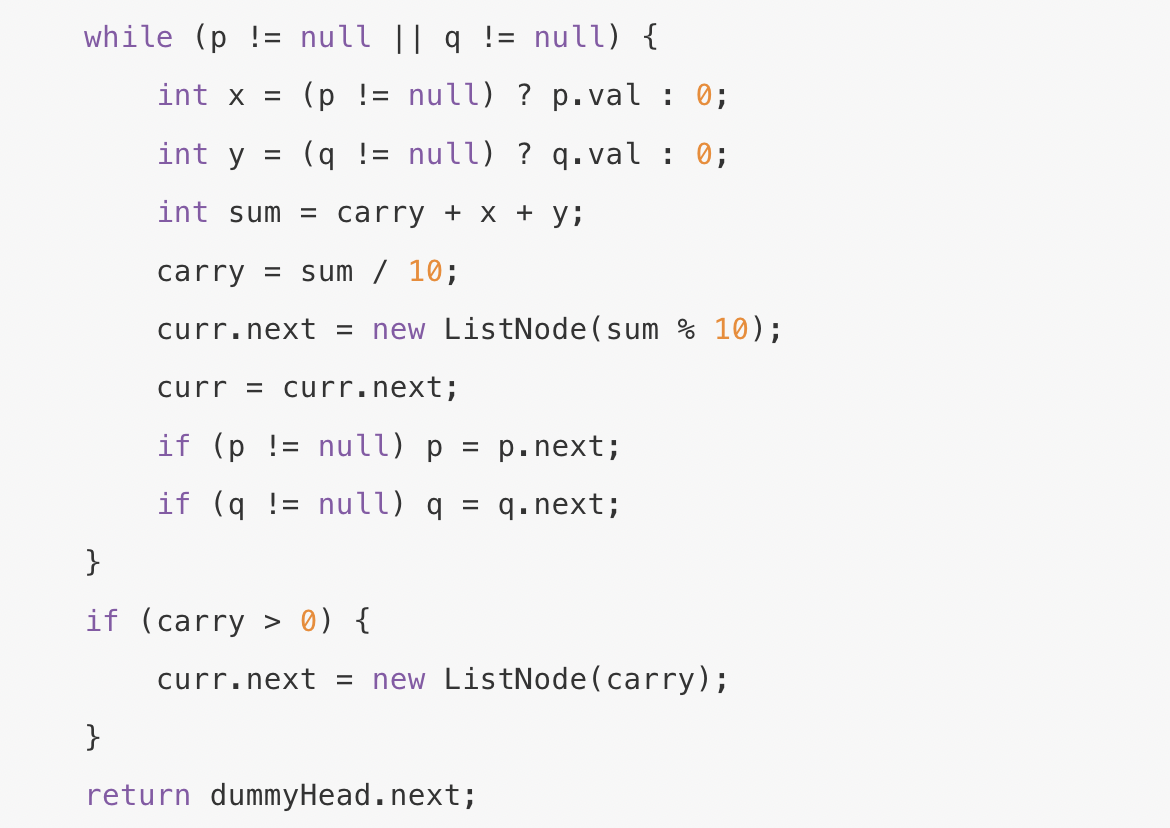
格雷码：当前数据异或最高位。

for(int binary = 0;binary < 1 << n; binary++){

gray.add(binary ^ binary >> 1);

}

两个链表相加技巧：将空的节点值赋值为0



最长字串问题：

HashSet保存之前的元素。挨个跳转至不重复元素为止。

HashMap当遇到重复元素时，计算距离重复元素距离。

数组，定一个长度为128的数组，以字母的 ASCII为下标。

当插入j个元素时遇到重复元素，再次访问都到该位置，记住需要跳转的位置。

