



吴军

今天 00:00



第117封信 | 语音和图像压缩..

9:14 8.46 MB



信件朗读者：宝木

小师弟，你好！

这一周我们介绍计算机科学的一些原理，这些内容虽然可能有点专业化，但是我会讲解得比较简单，因此大家不要有顾虑。

我们还是从几道Google的面试题入手，每一道题会引出背后的一个原理。今天我们要讲的面试题是倒置英语句子，这也是我在面试Google时被考到的一个问题。

问题是这么说的：

给你一个英语的语句，比如"London bridge is falling down"，把它完全倒装过来，"down falling is bridge London"，如何不使用额外的存储空间完成这个倒装过程？

通常学习计算机算法的人在解决这个问题时，首先会想到把这个句子切割成一个个单词，然后把它们存到一个数组里，数组的结构大致如下：

把这个数组顺序存入，逆序取出来就可以完成语句倒装的问题。当然，还有一个类似的办法，就是把上面的单词，一个个送入堆栈，如果你还记得[我写给你的第98封信](#)中，介绍堆栈的先进后出，后进先出性质时，就可以利用这个数据结构完成句子的倒装。

但是，这种算法要额外地使用存储空间，因此不符合题目的要求。在面试时，我们一般会让选择了上述方法的候选人把他们的想法说完，这样至少让他们在心理上不至于感受到打击，但是接下来我们会要求他们找出不使用额外内存空间的方法。

很多人想到的是把上面句子中的单词前后对调。但这道题目的难点恰恰在于英语单词的长度不同，如果不使用额外的空间，很难把不同长度的单词对调。

学习计算机的人会想到记录下来句子一头一尾两个单词的长度，然后把长的那个单词先挪开，短的那个填进长的单词空出来的位置。

比如在上面的例子中，London这个词比较长，down这个词比较短，可以把London先挪出来，把down这个词放到London的位置中，这是放得下的。但是这样接下来的问题又来了，长的单词London无法填入短的单词down留出来的空位。

当然，有人会想，在短的单词那边再挪走一个词，具体到上面的例子中，就是挪走falling，看看能否把长的单词安置进去。在这个例子中是可以的。当然，实际情况可能会比这个复杂，有可能留出的空间还不够，比如of the 这两个单词的长度加起来也没有Chinese一个长。即便句子尾巴上两个单词的位置能够放头上一个长的单词，但也有可能挪出的空间太多了，这样句子的头上放不下两个单词，上面的例子就陷入了后一种情况。

我讲到这里，你可能已经糊涂了，这其实就对了，因为我也没有打算让你听懂，我啰哩啰嗦地讲了这么多，无非是想说，前后对调单词这件事远比我们想象的复杂得多，以至于这个看似合理的想法在计算机中几乎无法实现。

在我考察Google候选人中，我大约向十几个候选人问了这个问题，大约一半的人（10人左右）试图采用头尾单词对调的方法，但是这10个人没有一个人能够写出实现他们想法的程序，因为他们都走进了死胡同。

上面这种方法的问题在哪里呢？其实最大的问题在于人会陷入自己固有的思维方式，或者说常人的思维。这道题目中，我们要求以单词为单元进行倒装，单词本身必须维持原状，因此大家为了满足这一点要求，不敢把单词打碎，搞乱。而用计算机解决这个问题关键，又恰恰是要把单词先搞乱，再规整起来。这道题最好的解法其实非常简单，如果你已经想到了，恭喜你，如果还没有，不妨花一分钟听我讲一下，其实只需要捅破一层窗户纸。

第一步，先将整个句子看成是一个完整的字符串，以字母为单位头尾对调，这样上面的句子就变成了下面这样一个乱七八糟的字符串：

"nwod gnillaf si egdirb nodnoL"

上面这一串字，你可能根本看不懂，但是没有关系。接下来我们再完成第二步，你就看清楚了。

第二步，把用空格分割的每一个字串以字母