串

时间限制：1.0s　　内存限制：256.0MB

问题描述

给你一个串 s[0~n-1],要求你选择两个数 i,j,满足 0<=i<=j<=n,然后将 s[0~i-1]、s[i~j-1]、s[j~n-1]翻转，要求翻转后的串字典序最小。

输入格式

本题有多组数据，第一行一个数 T，表示数据组数。

每组数据占一行，为一个串。

输出格式

对于每组数据输出两个数 i、j，即变化后字典序最小的方案，多种方案任意输出一组方案即可。

样例输入

```
2
bacbadcba
abc
```

样例输出

```
2 5
1 2
```

数据规模和约定

串由小写字母构成；

令 S 为所有串长度之和；

对于 10%的数据，S<=300；

对于 30%的数据，S<=2000；

对于 60%的数据，S<=200000；

对于 100%的数据，S<=10000000；

请使用 gets 或者 scanf 或者更快的方法读入；

数据有梯度。

题解

【大暴力之术】

首先直接按照题解，我们可以马上得到一个最简单的算法：

枚举 i,j 直接更新答案，这里枚举 i,j 的复杂度为 $O(N^2)$，更新答案的复杂度为 $O(N)$，因此整个算法的复杂度为 $O(N^3)$，期望得分为 10。

【初步分析】

这里涉及到了翻转，所以我们不妨一开始就把整个序列翻转过来，这里题目意思就变成了这个样子：

给你一个串 s[0~n-1]，要求你选择两个数 i,j，满足 0<=i<=j<=n，然后将 s[0~i-1] 和 s[j~n-1] 交换，要求翻转后的串字典序最小。

这里我们发现题目相当于是要求找出一个后缀，把这个后缀提到最前面，然后对于剩下的一段，再继续找一个后缀提到前面。

假设我们先枚举提到最前面的后缀，会发现实际上我们只要将剩下的一段直接求一个环状的最小表示就可以了，一次，我们成功的将复杂度下降了一阶，只需要枚举 j，然后直接利用最小表示求出最优的 i 的位置即可，这里算法复杂度为 $O(N^2)$，期望的分为 30。

【深入分析】

这里的复杂度瓶颈变成了枚举 j 的位置，那么，有没有办法可以让我们快速的求出 j 的位置呢？

因为后缀 j 被提到了最前面，所以他相当于成为了第一关键字，那么，让后缀 j 字典序尽量小这是必然的，假设我们我们通过对所有后缀进行排序，得到了这样的一个序的结果：

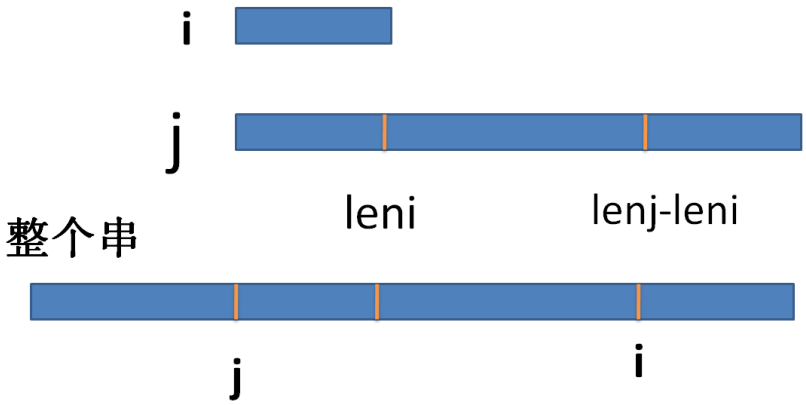
1. a
2. aba
3. ba

这里我们可以发现，第三个后缀是必然不优的，也就是说，如果后缀 i 和后缀 j 在还没有比较到结束符的时候就已经分出高下 i 更优，那么后缀 j 我们就没有必要去枚举了。

那么这里可以枚举的待选最优后缀集合里面，必然两两一个后缀是另外一个后缀的前缀，就如同上面的后缀 1 和后缀 2 之间的关系。

剩下的待选后缀集合可能仍然非常的大，可能达到 $O(N)$ 级别，所以仍然需要继续优化，这里，我们对于后缀 i 和后缀 j 到底谁被提到最前面更优，来讨论几种情况，假设后缀 i 的长度为 leni 后缀 j 的长度为 lenj：

Csse1: leni<lenj,leni*2<=lenj

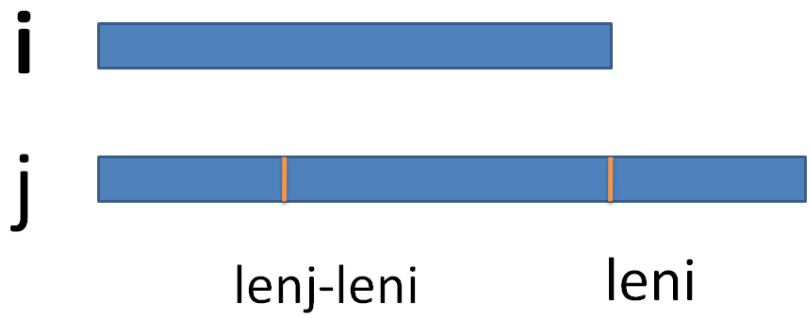


首先可以发现 si==sj[0,leni]==sj[lenj-leni,lenj]

这里我们先确定 si<sj[leni,lenj],那么我们总存在一种方案，可以使得选 i 要比选 j 更优，因为假设我将 i 提前以后，在剩下的那一段中选 j 为决策，那么这个串的开头将是 si+si，si+si>sj 因为我们确定了 si<sj[leni,lenj]。

Case2: leni<lenj,leni*2>lenj

在经过上面条件的筛选以后，剩下的后缀就是很有规律的了。



会发现 si==sj[0,leni]==sj[lenj-leni,lenj]。

那么 si 和 sj 必然都是以 gcd(leni,lenj)为循环的纯循环节，在这种情况下，必然是选 j 更优，因为如果我们选择将较短的提前，那么在接下来的最小表示中又会重新把 j 提前，这是浪费机会，结果是一样的，所以我们直接将 j 提前是较优的。

经过上面的分析以后，我们就可以得出算法了，我们首先找到这个串的字典序最小并且同时最短的串，然后在找到以他为纯循环节的最长的串即可。

因此，我们直接用一次最小表示求出 j 的位置，再用一次最小表示求出 i 的位置，整个算法复杂度为 O(N)，期望的分 100。

注：最小表示，求一个环状串的字典序最小的字符串的算法，见论文：

IOI2003冬令营演示文稿

安徽　周源

浅析“最小表示法”思想在字符串循环同构问题中的应用

安徽省芜湖市第一中学

周　源