

字符串加密

问题描述

H 国开发了一种新的字符串的加密方式，具体方法为将一个长度为 N 的由小写字母组成的字符串分成不超过 k 段，分别将每一段翻转，得到的新字符串即为加密以后的字符串。

H 国为 I 国的敌国，I 国为了打败 H 国，得到了一个加密以后的信息，我们想知道这个字符串在加密以前的样子，不过可能有多个答案，我们只要知道字典序最小的答案就可以了。

输入格式

第一行一个数 T ，表示数据组数。

对于每组数据，第一行两个数 N, K 分别表示字符串的长度和被分成了 K 段。

第二行一个长度为 N 的字符串。

输出格式

对于每组数据，输出一个长度为 N 的字符串，即加密前的可能的字典序最小的字符串。

样例输入

```
1
1 3
bacbadcba
```

样例输出

```
ababcabcd
```

数据规模和约定

测试点编号	数据组数 T	字符串长度 N 的规模	分裂段数 k 的规模
1 ~ 2	$T \leq 10$	$1 \leq N \leq 300$	$1 \leq K \leq 3$
3 ~ 6		$1 \leq N \leq 2000$	
7 ~ 10		$1 \leq N \leq 100000$	
11			$K = 1$
12			$K = 2$
13 ~ 16			$1 \leq K \leq 100$
17 ~ 20		$1 \leq K \leq N$	

题目简述

给定一个长度为 N 的字符串，要求将其分成不超过 K 段，并将分成的每一段进行翻转操作，使得到的串的字典序最小。

算法 1

【分析】

观察测试点 11 的数据范围，我们可以发现 $K=1$ ，也就意味着，题目要求我们将这个字符串分为一段翻转，这一段必然就是整个字符串。

【时间空间复杂度】

$O(N)$

【期望得分】

5 分

算法 2

【分析】

通过观察范围，可以大量的数据 $K \leq 3$ ，题目要求我们分成小于等于 3 段，那么意味着这个字符串会有两个断点，我们暴力枚举断点即可。

【时间复杂度】

$O(C_N^{K-1} * N)$

【期望得分】

15 分

算法 3

【分析】

观察测试点 12 的数据范围，在这个测试点中 $N \leq 100000, K \leq 2$ ，如果仍然使用算法 2 来解决，时间复杂度将为 N^2 ，无法通过此测试点。

解决此测试点，我们不妨换一个思路，一开始就将整个串翻转
`bacbadc => abcdabc`

我们会发现，我们的任务变成了，将序列分成 2 段，并将这 2 段交换位置。到这里我们会发现，这个问题变成了经典的最小表示问题。至此我们通过将其转化为最小表示问题，就可以解决测试点 12。

在这个算法的启发下，我们可以通过枚举断点到最后只剩下两段的时候使用最小表示。

【时间复杂度】

$$O(C_N^{K-2} * N)$$

【期望得分】

40 分

算法 4

【分析】

仍然是引用算法 3 的思想，我们一开始就将整个串翻转，这时候我们的任务变成了将这个字符串分成 K 段，将这 K 段变成逆序，如下：

agafaeadacab

abacadaeafag

由于要求最后的字典序最小，所以这里相当于我们一最后一段为第一关键字，倒数第二段为第二关键字，以此类推。

那么我们来分析下到底应该以什么策略来进行贪心：

假设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：

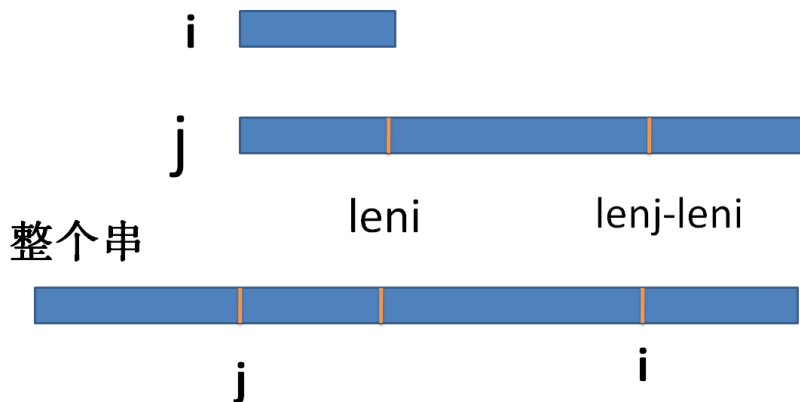
1. a
2. aba
3. ba

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 len_i 后缀 j 的长度为 len_j ：

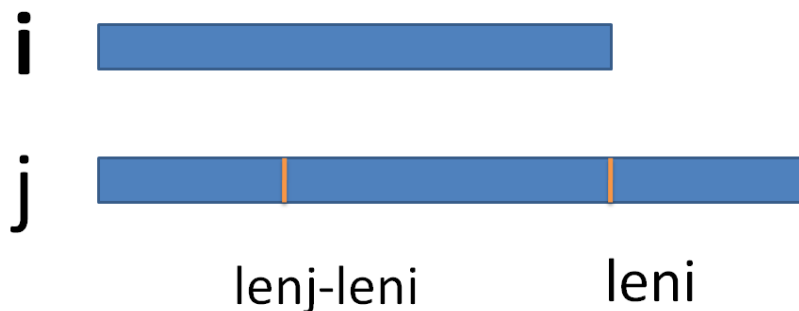
Cs1: $len_i < len_j, len_i * 2 \leq len_j$



首先可以发现 $s_i == s_j[0, len_i] == s_j[len_j - len_i, len_j]$
 这里我们先确定 $s_i < s_j[len_i, len_j]$, 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 $s_i + s_i$, $s_i + s_i < s_j$ 因为我们确定了 $s_i < s_j[len_i, len_j]$ 。

Case2: $len_i < len_j, len_i * 2 > len_j$

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 $s_i == s_j[0, len_i] == s_j[len_j - len_i, len_j]$ 。
 那么 s_i 和 s_j 必然都是以 $\gcd(len_i, len_j)$ 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, 我们每次用 $O(N)$ 的复杂度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。

【时间复杂度】

$O(NK)$

【期望得分】

80 分

算法 5

【分析】

每次为了快速的找到最优后缀的位置，我们可以用数据结构优化，可以完成这项工作的数据结构很多：后缀数组、后缀树、后缀自动机等等都可以胜任，这里不再做过多的介绍。

【时间复杂度】

$O(N) \sim O(N \log N)$

【期望得分】

100 分