Mr. Kitayuta's Gift 解题报告

绍兴市第一中学 王鉴浩

1 试题来源

CF 506E

2 试题大意

给一个只包含小写英文字母的字符串 s。

现在每次要在这个串中添加一个小写英文字符，字符可以添加在串的任何位置（包括头尾）。经过严格添加 n 个字符后，我们得到了一个新的字符串，要求这个串是回文串。

现在询问，最后能得到几个不同的的字符串，答案模 10007。

对于 100% 的数据：$1 \le |s| \le 100, 1 \le n \le 10^9$

时限：$6s$

3 算法介绍

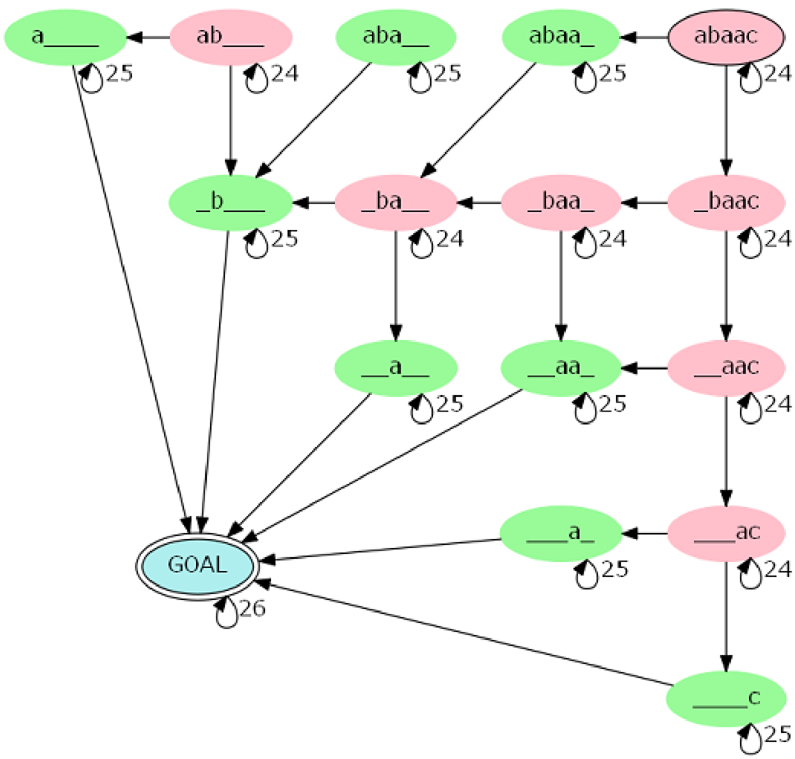
题目等于是找多少个长 $n + |s|$ 的回文串，要求 s 是这些串的子序列。由于是回文串，所以，我们可以把串切半，只对于半个串进行分析。

我们可以运用动态规划来解决此题。

首先，可以用 $f[i][j][k]$ 表示匹配了 i 个字符，s 串左半匹配到了第 j 位，右

半匹配到了第 k 位。对于这个动态规划的转移方程，我们用矩阵乘法来加速，可以在时间复杂度为 $O(|s|^6 * \log n)$ 内解决。

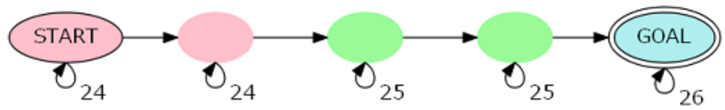
然后我们来优化这个转移。对于每次转移，根据目前添加的字符和目前匹配的位置，会有不同类型的转移。在下图中，我们把状态和转移画出来，其中 s = abaac。图中绿色的状态表示左半第 j 位和右半第 k 位的字符相等，粉色的状态反之。所以这两个状态的转移不一样。



于是，我们可以发现，状态其实只有 3 种：粉色，绿色和目标状态；每种状态本质是一样的。而且，对于绿色的状态，如果匹配的话，我们可以理解成在 s 串上走了 2 步；粉色的状态可以理解成走了 1 步。于是，就可以枚举走过了多少个绿色的状态，因为总步数一定，那么粉色的状态的个数也固定了。其中，当固定绿色和粉色状态个数后，我们可以通过动态规划计算出方案数。我们可以在总时间复杂度为 $O(|s|^3)$ 内完成上述动态规划计算。

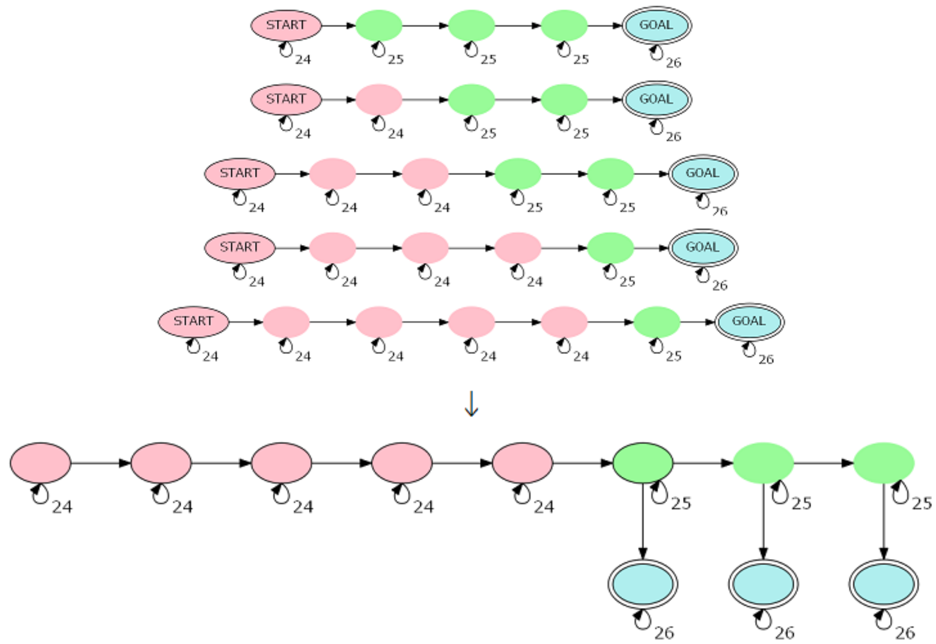
于是，我们现在只需考虑要走 x 个粉色状态和走 y 个绿色状态的方案数，由于顺序没有关系，所以可以认为是先走 x 个粉色状态再走 y 个绿色状态再到

终点，转移如下图：



由于我们要对每个数量的绿色状态都计算一遍，所以运用矩阵乘法加速后，这个解法总时间复杂度为 $O(|s|^4 * \log n)$。

我们需要继续优化我们的转移。仔细分析的话，是可以发现对于不同个数的绿色状态，它们的转移矩阵是可以合并的。我们就可以得到一张有 $2|s|$ 个状态的转移，具体合并如下图：



于是，我们就可以经过一次计算，得到每个绿色状态数量的转移答案。这个解法的时间复杂度为 $O(|s|^3 * \log n)$。