

字符串浅谈

by 冯青同

CF25E Test

- 直接枚举 s_1, s_2, s_3 的6种排列，然后利用哈希判断相邻串的公共前后缀长度即可。
- 注意要考虑相互包含的情况。时间复杂度 $O(N)$

CF1056E Check Transcription

- 考虑枚举 0 串的长度，然后可以相应的计算出 1 串的长度，利用哈希check是否可行即可。
- 单次check的时间复杂度是 $O(\text{len}T)$ 的，check的次数上课的时候我证明了（但我还没hack那个题解），涉及到一部分数论知识，这里不做赘述。

P7114 字符串匹配

- 首先枚举 AB 串为 T，然后考虑直接暴力判断 $(AB)^i$ 是否可行，然后 C 就已经确定了，然后只需要统计 T 串的划分中能使 A 中出现奇数次的字符个数大于 C 中出现奇数次的字符个数即可。
- 这里分析复杂度时利用到了调和级数的知识，复杂度为 $O(n \log n)$

P2870 Best Cow Line G

- 直接贪心的话，每次在左右两端取较小的那个即可。
- 但是还有个问题，那就是假如左右两端的字符相同应该怎么办？
- 那考虑从左右两端向内扫，找到第一个不同的字符，先取这个不同的字符较小的那一端，答案一定不会更差。
- 直接扫的复杂度可能会被卡到 $O(N^2)$ ，这里使用二分加哈希加速可以做到 $O(N \log N)$

P6286 [COCI2016-2017#1] Cezar

- 考虑对于相邻的两个串 $s(A_i)$ 和 $s(A_{i+1})$ 因为 A_i 的排名最靠前，所以只需要找到 A_i 和 A_{i+1} 第一个不同的字符，分别设为 x_i 和 x_{i+1} ，那么就是 x_i 的字典序一定要比 x_{i+1} 小。
- 这个过程可以使用二分+哈希优化。
- 这样处理完以后，会得到若干个字符之间的不等关系，然后只需要跑一遍拓扑排序即可。

P5829 【模板】失配树

- 先了解 KMP 的 border 的概念，然后再来做这道题。
- 因为我的 border 的 border 还是我的 border，所以只需要 p 和 q 一直跳 border，直到它们二者的 border 相同即可。
- 考虑令 i 的父亲为 fi 建一颗树，这棵树就叫失配树，那么上面的过程就可以转化为求 lca 了。

P2375 [NOI2014] 动物园

- 不断跳 border, 直到 border 的长度小于 $i/2$ 即可。
- 这个过程有点慢, 是 n^2 的。
- 使用倍增优化可以做到 $n\log n$ 。
- 正解是 $O(n)$ 的, 具体来说, 先预处理 fi 数组, 然后在求 num_i 的时候直接用类似 kmp 的方法转移即可。
- 证明的话, 就是 $i+1$ 的答案一定是由 i 的答案扩展而来的, 因为假如 $i+1$ 的答案用了一个大于 $i/2$ 的字符串来扩展, 那么得到的答案一定比 $(i+1)/2$ 大。

UVA11732 "strcmp()" Anyone?

- 这是一道字符串的计数题。
- 首先考虑只有两个字符串，设他们的最长公共前缀长度为 len ，则比较次数为 $2*len+1$
- 特别的，假如两个字符串完全相同，那么比较次数是 $2*len+2$
- 假设我们把前 i 个单词建了一个字典树，然后对第 $i+1$ 个串统计答案，操作是这样滴：
- 1、第 $i+1$ 个串没到结尾，假设当前扫到了第 j 个字符，在 u 节点，则 $ans+=(sum[u]-sum[son[u][ch]])*(2*j+1)$
- 2、第 $i+1$ 个串到结尾了， $ans+=(sum[u]-ed[u])*(2*j+1)+ed[u]*2*j$
- sum_u 表示经过 u 节点的串的个数， ed_u 表示在 u 处结尾的串的个数。

P4421 [COCI2017-2018#1] Lozinke

- 统计有多少个形如 A 是 B 的子串的关系。
- 那就考虑把所有字符串按照长度从大到小排序，然后每读进一个字符串就先判断一下它本身有没有出现过，再把它的所有子串的哈希值加到map里。
- 也可以用trie树来做，按长度从小到大排序，统计路径上的edu的总和即可。
- 注意要特判两个字符串完全相同的情况。

P2922 [USACO08DEC] Secret Message G

- 简单的trie树统计题。
- 读入一个串更新答案时，统计沿途的edu和结尾处的sumu即可。
- 备注：edu和sumu的定义均与前面完全相同。