

后缀数组的应用

寻找最小的循环移动位置

将字符串 S 复制一份变成 SS 就转化成了后缀排序问题。

在字符串中找子串

任务是在线地在主串 T 中寻找模式串 S 。在线的意思是，我们已经预先知道主串 T ，但是当且仅当询问时才知道模式串 S 。我们可以先构造出 T 的后缀数组，然后查找子串 S 。若子串 S 在 T 中出现，它必定是 T 的一些后缀的前缀。因为我们已经将所有后缀排序了，我们可以通过在 p 数组中二分 S 来实现。比较子串 S 和当前后缀的时间复杂度为 $O(|S|)$ ，因此找子串的时间复杂度为 $O(|S| \log |T|)$ 。注意，如果该子串在 T 中出现了多次，每次出现都是在 p 数组中相邻的。因此出现次数可以通过再次二分找到，输出每次出现的位置也很轻松。

从字符串首尾取字符最小化字典序

题意：给你一个字符串，每次从首或尾取一个字符组成字符串，问所有能够组成的字符串中字典序最小的一个。

暴力做法就是每次最坏 $O(n)$ 地判断当前应该取首还是尾（即比较取首得到的字符串与取尾得到的反串的大小），只需优化这一判断过程即可。

由于需要在原串后缀与反串后缀构成的集合内比较大小，可以将反串拼接在原串后，并在中间加上一个没出现过的字符（如 $\#$ ，代码中可以直接使用空字符），求后缀数组，即可 $O(1)$ 完成这一判断。

height 数组

LCP（最长公共前缀）

两个字符串 S 和 T 的 LCP 就是最大的 $x (x \leq \min(|S|, |T|))$ 使得 $S_i = T_i (\forall 1 \leq i \leq x)$ 。

下文中以 $lcp(i, j)$ 表示后缀 i 和后缀 j 的最长公共前缀（的长度）。

$O(n)$ 求 height 数组的代码实现

利用上面这个引理暴力求即可：

```
1 for (i = 1, k = 0; i <= n; ++i) {
2     if (rk[i] == 0) continue;
3     if (k) --k;
4     while (s[i + k] == s[sa[rk[i] - 1] + k]) ++k;
5     height[rk[i]] = k;
6 }
```

k 不会超过 n ，最多减 n 次，所以最多加 $2n$ 次，总复杂度就是 $O(n)$ 。

height 数组的应用

两子串最长公共前缀

$$lcp(sa[i], sa[j]) = \min\{height[i + 1..j]\}$$

感性理解：如果 $height$ 一直大于某个数，前这么多位就一直没变过；反之，由于后缀已经排好序了，不可能变了之后变回来。

严格证明可以参考[[2004] 后缀数组 by. 徐智磊][1]。

有了这个定理，求两子串最长公共前缀就转化为了 [RMQ 问题](#)。

比较一个字符串的两个子串的大小关系

假设需要比较的是 $A = S[a..b]$ 和 $B = S[c..d]$ 的大小关系。

若 $lcp(a, c) \geq \min(|A|, |B|)$, $A < B \iff |A| < |B|$ 。

否则, $A < B \iff rk[a] < rk[c]$ 。

不同子串的数目

子串就是后缀的前缀，所以可以枚举每个后缀，计算前缀总数，再减掉重复。

「前缀总数」其实就是子串个数，为 $n(n + 1)/2$ 。

如果按后缀排序的顺序枚举后缀，每次新增的子串就是除了与上一个后缀的 LCP 剩下的前缀。这些前缀一定是新增的，否则会破坏 $lcp(sa[i], sa[j]) = \min\{height[i + 1..j]\}$ 的性质。只有这些前缀是新增的，因为 LCP 部分在枚举上一个前缀时计算过了。

所以答案为：

$$\frac{n(n+1)}{2} - \sum_{i=2}^n height[i]$$

出现至少 k 次的子串的最大长度

出现至少 k 次意味着后缀排序后有至少连续 k 个后缀以这个子串作为公共前缀。

所以，求出每相邻 $k - 1$ 个 $height$ 的最小值，再求这些最小值的最大值就是答案。

可以使用单调队列 $O(n)$ 解决，但使用其它方式也足以 AC。

是否有某字符串在文本串中至少不重叠地出现了两次

可以二分目标串的长度 $|s|$ ，将 h 数组划分成若干个连续 LCP 大于等于 $|s|$ 的段，利用 RMQ 对每个段求其中出现的数中最大和最小的下标，若这两个下标的距离满足条件，则一定有长度为 $|s|$ 的字符串不重叠地出现了两次。

连续的若干个相同子串

我们可以枚举连续串的长度 $|s|$ ，按照 $|s|$ 对整个串进行分块，对相邻两块的块首进行 LCP 与 LCS 查询，具体可见[[2009] 后缀数组——处理字符串的有力工具][2]。

结合并查集

某些题目求解时要求你将后缀数组划分成若干个连续 LCP 长度大于等于某一值的段，亦即将 h 数组划分成若干个连续最小值大于等于某一值的段并统计每一段的答案。如果有多次询问，我们可以将询问离线。观察到当给定值单调递减的时候，满足条件的区间个数总是越来越少，而新区间都是两个或多个原区间相连所得，且新区间中不包含在原区间内的部分的 h 值都为减少到的这个值。我们只需要维护一个并查集，每次合并相邻的两个区间，并维护统计信息即可。

结合线段树

某些题目让你求满足条件的前若干个数的，而这些数又在后缀排序中的一个区间内。这时我们可以用归并排序的性质来合并两个结点的信息，利用线段树维护和查询区间答案。