【后缀数组】 - 沐阳 - 博客园

后缀数组是解决一系列字符串题目的利器，后缀数组中保留了这样的信息。sa[i]表示排名为第 i 位的后缀是从sa[i]开始的。通过倍增算法可以在O(nlogn)的时间复杂度内将所有的后缀进行排序。而height数组也是在处理问题中经常要使用到的，height[i]表示排名第 i 的后缀与排名第 i-1 位的后缀的最长公共前缀的长度。具体见代码。

⊞

模板-注释

⊞

模板-清爽

PS:后面的题目，da函数中均写错了一个地方，即统计ws数组的时候，i 应该是从1开始的，模板已改正，后面的就不改了.....

重复子串：字符串R在字符串L中至少出现两次，则称R是L的重复子串

1）可重复性最长重复子串：求出height数组最大的一个值即可。

2）不可重复最长重复子串：二分枚举长度，将原问题转化为一个判定性的问题，

若枚举的长度为k那么根据height值进行分组，连续的height值大于等于k的分为一组，通过查看一组内部的最大后缀编号和最小后缀编号差值是否大于等于k来判定。

题意：给定若干个数字，其大小在1-88之间，现在要出要出这个串中最长的不重复的两个相同的子串，满足要求的子串的定义是长度不短于5，其两个子串可以通过分别加上某一值或减去某一值得到，串不能够有公共部分。分析：首先通过该串的后一项减去前一项得到一个反应差值的新序列，然后再对这个新序列求sa数组即height数组，使用二分答案的方式来解决这个问题，需要注意的是转化为差值之后，在计算编号差的时候不能够取等号，否则将会使得共用一个元素的情况发生。

⊞

View Code

3）可重叠的K次最长重复子串：要求串中该子串至少出现K次，但是某一段字符串可以贡献于多个子串。做法是二分枚举长度，然后与第2种一样进行分组，不同的这里需要统计同一组的后缀个数是否大于等于K即可。

题意：给定一系列的数字串，求出这其中出现次数不低于K次的最长的子串，题目保证有结果。分析：由于题目中给定的数字集合较大因先对数字进行离散化处理。然后通过后缀数组求出sa和height数组后，二分枚举长度进行分组，统计一组内的后缀个数是否大于K。

⊞

View Code

子串的个数：求出给定一个串中不相同子串的个数

1）不相同子串的个数：每个子串一定是某个后缀的前缀，那么原问题等价于求所有后缀之间的不相同的前缀的个数。对于每个后缀在考虑其带来的前缀的时候考虑其重复的部分减去就行了，重复前缀的数量

为height数组在该子串处的值，可以理解为所有后缀中与该后缀最相似的公共前缀部分不再予以统计。如果说还有其他串与该串有公共前缀，那么这个串一定会包含在与最相似的串的相同前缀中。总的来说是一个传递的问题，保证一个前缀被统计的一次就不再会被统计第二次。

题意：给定长度为N的字符串，求出其中不相同子串的个数，两题只是给定的N的大小不一样。分析：求出sa、height、rank数组后。一种方式是枚举排名，一种是枚举位置。代码选择是后一种。

⊞

View Code

2)字符串任意区间的子串个数：首先对整个字符串求一次sa[]以及height[]，之后对于任意区间[L, R]，遍历一遍sa[]，只要起点在[L, R]内的后缀就需要进行统计，类似于1）中的方法，不过有一个地方要特别注意的就是全部的sa[]不一定就是区间内的sa[]，这是因为区间内的后缀比较时有额外的长度限制。可以证明遍历的过程要遵循如下的规则：

后缀s1和后缀s2现在是两个待比较的后缀，s1在前，s2在后，其起点都在区间[L, R]内，并设两串在区间中的长度为 len1, len2, 其全局的最长公共前缀为 lcp。现考虑在遍历sa[]时，如何从全局sa[]得到正确的局部sa[]：

1: lcp < len1 && lcp < len2 时说明两个串在未结束时就比较出了大小，全局和局部的sa[]统一，因此可以放心令s2作为下一个字典序后缀；

2: lcp >= len1 && lcp >= len2 时说明在其中一个串结束时，两个串对应字符都是相等的，这时需要根据len1和len2关系来决定，如果len1>len2那么就不用更换了；

3: lcp >= len1 && lcp < len2 时说明在其中一个串结束时，两个串对应字符都是相等的，由于s2的长度比s1长，因此字典序肯定大，因此需要更换当前的后缀；

4: lcp < len1 && lcp >= len2 时说明在其中一个串结束时，两个串对应字符都是相等的，由于s1的长度比s2长，因此字典序肯定大，因此不需更换当前的后缀。

其中2和4条件可以合并，如果4成立，那么必定有 len1 > len2，因此可以简化这个判断这个过程：if (len1 > len2 && lcp >= len2) 则不更换 else 更换。

直接理解就是给结果带来误差的情况只会是某个被lcp完全包含的后缀被排在了后面，那么它的正确位置应该是在最前面，由于在后面匹配的其长度仍未整个局部后缀的长度，而在选择下一个字典序后缀时屏蔽掉这个后缀即可。

HDU-4622 Reincarnation题意：给定一个字符串，询问[L, R]之间的子串数量。分析：除了使用上述方法外，另一种方法是直接对任意区间求出一个hash值（矩阵），然后dp求解某一区间（子矩阵）不同子串的数量。在处理 rmq 时预处理出log函数大大加快了速度。

⊞

View Code

回文子串：字符串L的某个子字符R反过来写后和原先的字符串R一样

1）最长回文子串：将给定的字符串反序后添加到原串的末尾，中间用特殊字符隔开，设这个字符为$。设原串长度为N，枚举0-N的每一位作为中心（奇数回文串、偶数回文串中心偏左的字符），那么回文串的长度其实枚举位置开始的后缀和关于$对称位置后缀（奇数回文串、偶数回文串为对称位置向后移一位）的最长公共前缀。中间插入的字符不能够是添加到串尾的字符，其作用就是防止最长公共前缀超过一个串的原本的最长长度。

URAL-1297 Palindrome题意：给定一个字符串，求其中的最长的回文串。分析：解法如上所述，使用的树状数组来求区间最小值。其实使用Manacher算法能够很好解决该题。

View Code

连续重复子串：如果一个字符串L是有某个字符串S重复R次而得到，则称L是一个连续重复串。R是这个字符串的重复次数

1）连续重复子串：给定一个串L，一直这个字符串是由某个字符串S重复R次而得到的，求R的最大值。求出sa和height数组后，枚举重复的串的长度K（K为L的约数），判定只需求suffix(1)和suffix(1+k)的最长公共前缀是否等于n-k，其实也就等价于KMP算法的含义了，计算任意一个后缀与第一个后缀的最长公共前缀。

题意：给定一个字符串，已知其实一个字符串重复n次而来，求出最大的n。分析：按照上面的解法，倍增算法TLE，DC3勉强能过。下面贴个KMP的代码。

View Code

2)重复次数最多的连续重复子串：给定一个字符串，求出重复次数最多的连续重复子串。解决这个问题的过程不像前面那些问题那么直接了。按步骤来，首先求出sa数组及height数组，接着枚举连续重复子串中不断重复那部分字符串的长度，称这个串为元串，长度为 i ；从1到max(len/2, 1)，枚举的元串假设其重复次数至少为两次，那么其一定会跨过两个 i 的整数倍点，因为如果只跨过一个点的话，其最长长度为2*i-1，不满足重复至少两次的要求；对两个整数倍点 i*j, i*j+i 求一次lcp，得出的长度为该元串在 i*j 位置能够重复出现的最长长度L，将这个长度除以 i 加1将得到重复次数L/i+1。这还没完，因为经过这两个点的情况还不完备，应还可以假设起点在 [i*j-i+1, i*j-d]，其中 d = i-L/i 其意义为根据已知的匹配长度，可以将起点往前移动的范围，太靠后将不能够构造出比之前更好的解。如果要求出某个最多的连续重复子串的最小字典序子需要枚举所有起点，但如果只是要的到最多的重复次数或者任意最多的连续重复子串，那么只需要枚举i*j-d处的起点即可，因为后面的起点若能够得到最优的结果，那么i*j-d处也一定能得到，且答案一样均为L/i+2。

SPOJ-687 Repeats题意：给定一个字符串，求出重复次数最多的连续重复子串的重复次数。分析：直接求即可，每次只需要枚举除整数倍之外的一个起点。

View Code

题意：给定一个字符串，求出重复次数最多的连续重复子串，优先重复次数最多，否则输出字典序最小的子串。分析：与上题不同的是需要枚举多个起点，因此尽管答案相同，可能字典序较之要小。用到了string的一个赋值函数str.assign(const char *, int, int);

View Code

公共子串：如果字符串L同时出现在字符串A和字符串B中，则称字符串L是字符串A和字符串B的公共子串

1）最长公共子串：给定两个字符串A和B，求最长公共子串。首先将两个字符串合并成一个字符串，通过二分枚举长度，然后在height分组中查看是否存在在两个不同串的后缀即可。也可以证明最长的公共子串所在后缀是相邻的，因此可以直接遍历一遍height数组，判定相邻的两个height数组是否属于不同的两个串。

题意：给定两个串，求出最长的公共子串的长度。分析：使用上述的方法即可。

⊞

View Code

题意：较之上题，该题要求输出任意一个最长的公共子串，求解的时候保留一下子串的位置信息即可。

⊞

View Code

2）公共子串的个数：给定两个字符串A和B，求长度不小于K的公共子串的个数。将串B添加到A之后，中间使用一个特殊字符分隔开（为了防止两个后缀串的公共前缀跨越两个字符串），求出sa数组和height数组后，扫描一遍height数组，并且进行分组，分组的时候要维护一个height值单调上升的栈，栈中的每一个元素拥有两个属性，第一个是其值为多少，第二个是前面还有多少个能够提供这个值的一共有多少个（如果新加入的height值比之前较小时，将回收之前的height值，将其视为同一高度，直到遇到比它小的）。需要对height数组作两次。

题意：求两个串的公共子串的个数。分析：如上所述。

⊞

View Code

多个字符串的相关问题

1)不小于K个字符串中的最长子串：同时出现在不少于K个串的子串。做法同样二分枚举长度，分组后在一组中寻找出现字符串是否超过K即可。

题意：给定N个串，求超过一半串拥有的最长子串。分析：将所有的串连接起来后二分枚举长度分组。

⊞

View Code

2）每个字符串至少出现两次且不可重叠的最长子串：二分枚举长度后在同一分组中对每一个字符串保留一个最小的位置和一个最大的位置，最后查看是否每个串在同一组中都有至少两个后缀，并且后缀的坐标差大于枚举的长度。

题意：给定N个串，求每个串至少出现两次的最长子串。分析：如上所述。

⊞

View Code

<2016年3月>

最新评论

阅读排行榜

评论排行榜

推荐排行榜