全国信息学奥林匹克竞赛复赛模拟赛

字符串专题

解题报告

一、题目简介

本套题目为字符串专项训练，题目知识面涉及NOIP及更高难度。共4道题目，知识点各有不同，难度为递增趋势。其中前2道题为NOIP范围内，相对比较简单；后2道题为较NOIP有一定的提升，相对较难。

第一题《品茶》为基础题目，进行最基本的字符串处理即可；

第二题《归途》需要运用到KMP算法或其扩展算法；

第三题《钢琴手》的主体为AC自动机，计算时要用到动态规划，并且要写高精度运算；

第四题《幸福》用后缀数组找最长公共子串。

二、《品茶》解题报告

大概题意

给出 n 个字符串，求有多少个字符串中的Q/W/E/R个数超过该字符串的字符个数的1/2。

数据构造

数据点1-3： $1 \le n \le 15$，字符串长度 ≤ 30 ，其中：

数据点1：存在特殊字符，需要考虑；

数据点2：存在多个Q/W/E/R刚好为1/2；

数据点4-6： $1 \le n \le 500$，字符串长度 ≤ 1000 ，其中：

数据点4：对于每一个字符串均为相同字符；

数据点5：全部为大写字母；

数据点6：全部为小写字母；

数据点7-10： $1 \le n \le 2000$，字符串长度 ≤ 20000 ，其中：

数据点7-8：大小写字母混合；

数据点10：极限数据。

详细分析

由于我们要进行的操作就是查找字符串中的特殊字符，所以最简单有效的方法就是对于每一个字符串的每一个字符进行枚举，统计这个字符串中的Q/W/E/R个数，然后用该个数乘2看是否超过字符串总长度，求和即可。需要注意的是，由于大小写字母均存在，但是不区分，故需要进行两次判断或者进行大小写转化后再统计。时间复杂度为 $O(n * len)$ ，其中 len 为所有字符串长度的平均值。

但是我们注意到，对于100%的数据， $1 \le n \le 2000$，字符串长度 ≤ 20000 ，

一旦程序的常数写得过大，是很容易超时的。所以写程序的时候尽量养成良好的习惯，不去作一些浪费时间的无用功。

非完美算法和分数段分析

由于本题本身就是模拟，没有什么难度可言，真正丢分的只有可能因为没有看清题目，故不存在部分分算法。

三、《归途》解题报告

大概题意

给出一个长度为 n 的带权字符串和长度为 m 的子字符串，求子字符串是否出现在带权字符串中，且出现位置的最后一位之前的权值和是否超过限制权值，如果没有就输出该权值和。存在 x 组数据。

数据构造

数据点1：$1 \le n \le 5,\ 2 \le m \le 5$，并且带权字符串不存在环；

数据点2-3：$1 \le n \le 5,\ 2 \le m \le 20$ ，其中：

数据点3：权值全部为1；

数据点4-6：$1 \le n \le 20,\ 2 \le p \le m \le 5000$，均为随机数据；

数据点7-10： $1 \le x \le 10,\ 1 \le n \le 100,\ 2 \le p \le m \le 10^5,\ 1 \le t[i] \le 10^5,\ 1 \le num[i], u[i], v[i], a[i] \le n,\ \ 0 \le maxt \le \sum t \times 2$，其中：

数据点9-10： $maxt \ge MAXINT$，需要开 $long\ long$ 。

详细分析

题目表面上似乎是一道图论题， n 个节点， m 条边，并且还带权值，但是所询问内容令人费解：在图中找到一段路。如果题目要求每个节点只能访问一次，这样当然是简单的，但是由于不限制次数，故可能出现若干个环，并且对于每一条边，每次经过的权值都是不同的！如果直接用无向图来维护的话，基本上是不可能的。所以，我们可以从另外一个角度来考虑：我们将所走的这段路铺平，不将其视为图中遍历，而是看作一个线性数组，而题目要求为在这个数组中找到另一个线性数组。这样，题意就已经很明确了：子串匹配母串。（即大概题意所述）

确定了字符串匹配这个大主题，直接想到的就是暴力枚举，时间复杂度为 $O(x*n*m)$。根据题目所给数据范围，这显然是不能通过的。KMP算法可以很好地解决，时间复杂度为 $O(x*(n+m))$。当然扩展KMP算法也可行。

由于权值的存在以及对其的限制，我们只需要记录字符串中每个字符之前的权值之和，对于第一次成功匹配判断是否大于 $maxt$ ，如果是的，则说明无法匹配上；相反地，直接输出权值和。

要注意到的是，由于$2 \le p \le m \le 10^5,\ 1 \le t[i] \le 10^5, 0 \le maxt \le \sum t \times 2$，故超过 int 范围是不可避免的，所以需要开 $long\ long$ 。

非完美算法和分数段分析

对于10分的算法：由于不存在环，故可以构建无向图进行搜索，记录路径进行判断，时间复杂度略；

对于30-40分的算法：暴力枚举两个字符串的匹配位置，时间复杂度为 $O(x*n*m)$；

对于100分的算法：KMP算法进行匹配，时间复杂度为 $O(x*(n+m))$。

四、《钢琴手》解题报告

大概题意

给出一个带 n 个字符的字典，有 p 个禁用的单词，问能组成多少个不同的长度为 m 的字符串，不包括 p 个禁用的单词。如果不存在则输出-1。

数据构造

数据点1-6： $1 \le n \le 5,\ 1 \le m \le 3,\ 0 \le p \le 5$，其中：

数据点1-2：为不存在方案的情况，输出-1即可；

数据点4： $p = 0$，直接输出总方案数；

数据点7-8：$1 \le n \le 10,\ 1 \le m = len \le 10,\ 0 \le p \le 10$，其中 len 表示每一个有问题的片段长度；

数据点9-14： $1 \le n \le 20,\ 1 \le m \le 20,\ 0 \le p \le 10$，方案总数 $\ge MAXINT$，需要开 $long\ long$ ；

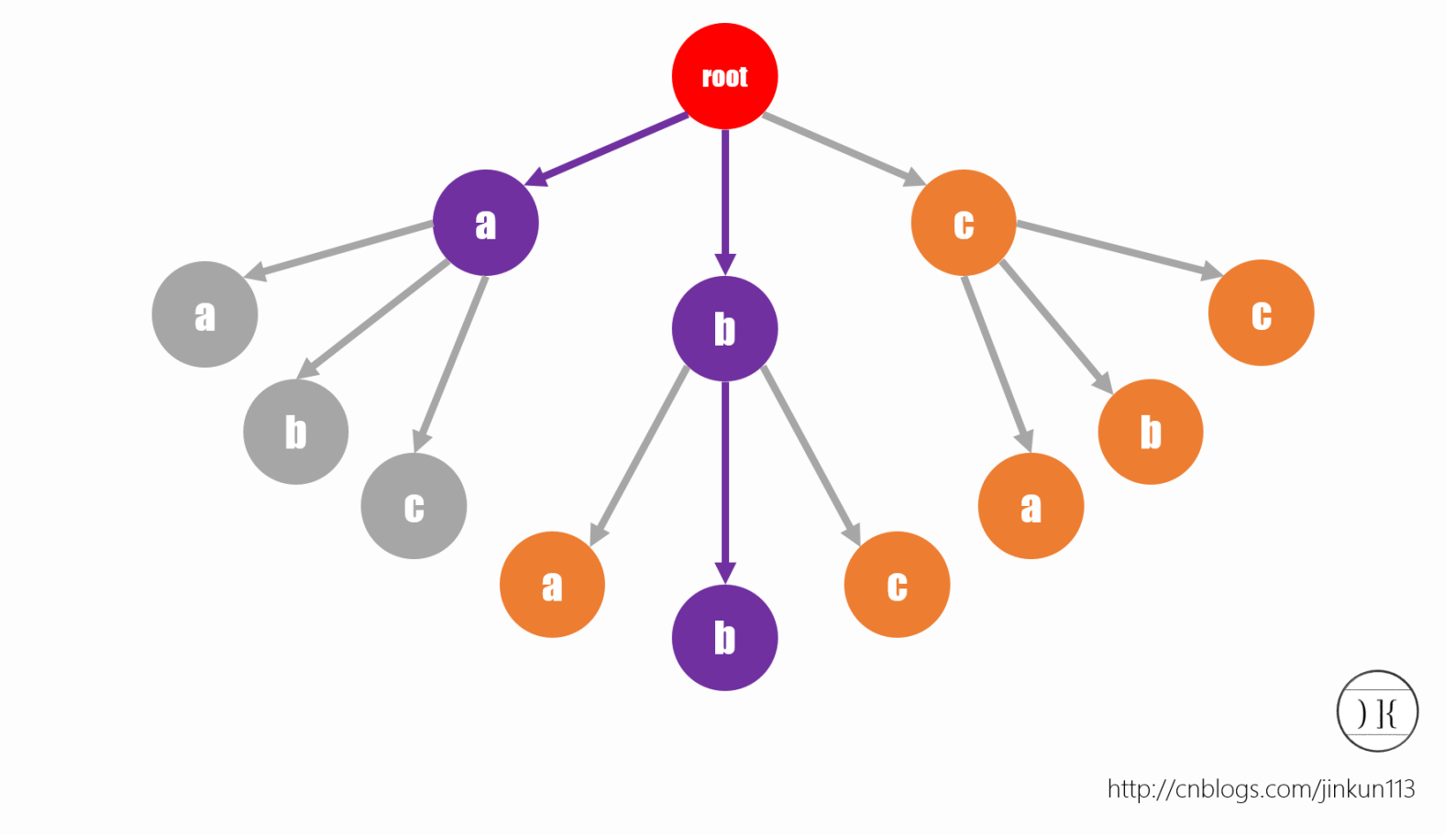
数据点15-20：$1 \le n \le 25,\ 1 \le m \le 50,\ 0 \le p \le 10$，方案总数 $\ge MAXLONGLONG$，需要高精度算法。

详细分析

n个字符，组成长度为 m 的字符串，假设我们无视那 k 个禁用单词的情况，那么情况为多少呢？其实就是个简单的数学问题，方案数为n^m。其实这样已经可以得到10分了，我们注意到，在数据点7-8中，由于每一个有问题的音乐片段长度和所弹奏长度相等，这样的话每禁止一个，方案数直接减1即可，即答案为 $n^m - k$ 。另外还有一个数据点 $p = 0$。

那么对于其他情况的话，不能简单的用公式来计算。首先可以确定的是，我

需要构建AC自动机，将题目要表达的意思形象化。因为单词的禁用，不仅影响到其本身，而且只要一段之中有这个单词，就都算不合法，那么我们将这 p 个禁用的单词放入AC自动机中，设单词的最后一个字符的节点为 i ，则节点 i 的所有儿子节点就都不合法了。这也同样意味着，对于两个字符串 a 和 b ，如果 a 是 b 的前缀，那么禁用 b 是可以忽略的（想一想，为什么）。如图所示，假设给出一个字符集合为{ a, b, c }的字典，禁用单词为： a, bb 。则构建的AC自动机如下：



紫色表示插入AC自动机的禁用单词，灰色和紫色部分都是不可选的，而剩下的橙色的个数之和就是方案数了。

然而接下来就是如何计算方案数了。不难看出，我们需要记录下来以进行统计的，只有字母本身，而不需要涉及到AC自动机中的节点的位置。所以可以用简单方便的动态规划，用 $f[i][j]$ 表示当前长度为 i ，字符为 j 的方案数。状态转移很简单： $f[i][j] = \sum f[i-1][k]$ ， k 表示能够在AC自动机中走到字符为 j 的节点的节点。

一切似乎都已经准备就绪，但是我们再来看一眼数据范围， 当 $n = 25$ ，

$m = 50$ 时，答案最大可能高达$25^{50} = 7.89e + 69$，所以最后计算的时候，需要进行高精度运算。

非完美算法和分数段分析

对于10分的算法：输出 -1 即可；

对于15分的算法：如详细分析中所言，只要求得 $n^m - k$ 即可，但是其中的第2个数据点需要使用 $long\ long$ ；

对于30分的算法：暴力求出所有可能性，然后枚举所有被禁用单词，逐一排除；

对于40-45分的算法：10-15分算法+30分算法；

对于50-60分的算法：如详细分析所讲，AC自动机构建不可行单词，动态规划维护；

对于100分的算法：在50-60分的基础上，使用高精度计算。

五、《幸福》解题报告

大概题意

给出 n 个字符串，找出最长公共子串，要求在每个字符串中都出现至少两次，并且没有重叠。

数据构造

数据点1-6：$1 \le n \le 5,\ 2 \le len \le 20$，其中：

数据点2：全字符串字符相同；

数据点9-14：$1 \le n \le 10,\ 2 \le len \le 200$

数据点15-20：$1 \le t \le 10,\ 1 \le n \le 10,\ 2 \le len \le 10000$，其中：

数据点19：全字符串字符相同；

数据点20：全英文文章。

详细分析

首先可以确定的是，单串匹配多串首选后缀数组。相比直接找最长公共子串，这道题附加的条件在于每个字符串内都要出现多次，且不重叠，那么暂时先不管，考虑如何求最长公共子串。

因为需要的是在每个字符串中都出现，我们显然不可能为每一个字符串都开一个后缀数组，这样操作麻烦，又浪费空间。可以考虑将所有的字符串全部连在一起，中间用特殊符号中断以区分。但是题意中写明了所有可见的字符都有可能出现，我们不能保证哪一个特殊符号不会在字符串中出现，所以可以考虑用一个标号数组进行区分，即 $num[i]$表示第 i 个后缀在第 $num[i]$ 个字符串中。

利用后缀数组，我们将每个后缀的名次计算出来，然后再计算相邻后缀的最长公共前缀，即$height[i]$。

现在考虑每个字符串内部需要出现多次的条件。其实在看明白了前面如何衔接不同字符串这个步骤之后，这就很好去写了，即将原来字符串之间的最长公

共前缀修改成每个字符串内部的最长公共前缀。求解的过程用二分答案来完成，每次记录所有字符串内部一个位置的最大值和一个位置的最小值，观察差值，求出所有答案的最大值即可。

非完美算法和分数段分析

对于0-30分的算法：搜索，暴力查找出现多次的子串（分数取决于你的常数）；
对于100分的算法：后缀数组。
虽然本题给出了部分分，但是其实得部分分的难度可能不亚于写正解，欢迎大家一起研究。