

HDOJ 2222【基础】

题目大意：

给出 N 个单词 ($1 \leq N \leq 10000$)，每个单词的长度不超过 50 个字符，然后给出一篇文章（长度不超过 100 万个字符），问一共有多少个给出的单词在文章中出现了。

输入：

第一行有一个整数 T ，表示有 T 组测试数据。

每一组测试数据第一行有一个整数 N ，接下来有 N 行，每一行有一个单词。最后有一行，给出了要查找的文章。注意，文章中可能重复出现某些单词。

输出：

每一组测试数据输出一行，包含一个整数，即有多少个单词在文章中出现了（而不是单词出现的合计次数）。

题解：

AC 自动机的模板题。关于 AC 自动机的原理，参见

<http://hi.baidu.com/nialv7/item/celce015d44a6ba7feded52d>。我的模板是由 <http://blog.csdn.net/niushuai666/article/details/7002736> 改写而来的，加入了手动内存管理（因为原代码是会泄漏内存的，虽然实际测试显示泄漏得不多）。

POJ 1204【基础】

题目大意：

给出一个 row 行 col 列的大写字母矩阵，给出 N 个长度不超过 100 个字符的单词，已知这些单词一定在字母矩阵中出现过，他们的方向可以是 8 个方位（上 0，右上 1，右 2……左上 7）。这些单词可能出现不止一次。现在要你找出每个单词出现的一个位置，以及其出现的方向。 $1 \leq row, col, N \leq 1000$ 。

输入：

第一行有三个空格整数 row 、 col 和 N 。接下来有一个 row 行 col 列的大写字母矩阵。最后有 N 行，每一行有一个单词。

输出：

N 行，按输入单词的顺序，每一行输出两个整数，即这个单词起始位置的行和列。行和列从 0 开始算。然后再输出一个大写字母表示单词延伸方向，0 对应 A，1 对应 B……7 对应 H。

题解：

将所有给出的单词建 AC 自动机，然后以字母矩阵的两个边框作为起点，左侧可以向右、上、右、右下延伸，右侧可以向左、上、左、左下延伸，上侧可以向左、下、下、右下延伸，下侧可以向左、上、右上延伸，一直延伸到另一侧边界，得到一个字符串作为文本串去进行匹配。注意记录一下当前文本串的起始位置和方向，这样匹配到一个单词结尾的时候可以算出起始位置。

HDOJ 3065 【基础】

题目大意、输入输出要求见原题中文。

题解：

基本上就是裸的 AC 自动机模板。在匹配的时候超出大写字母范围的直接指向 root 就好了。

POJ 3691 【中等】

题目大意：

给出 N ($1 \leq N \leq 50$) 个长度不超过 20 个碱基的致病 DNA 片段，再给出一段长度不超过 1000 个碱基的目标 DNA 片段，求最少需要改变多少个目标 DNA 片段上的碱基，才能使其不含任何致病 DNA 片段。

输入：

有若干组测试数据。每一组测试数据第一行有一个整数 N，N=0 时测试数据结束。接下来有 N 行，每一行有一个致病 DNA 片段（由 ACTG 四个大写英文字母组成），然后再加一行目标 DNA 片段。

输出：

每一组测试数据输出一行，格式为：Case %测试数据组编号%: %最小碱基改变数%。其中测试数据组编号从 1 开始，如果无论如何都避不开致病 DNA 片段，最小碱基改变数输出-1。

题解：

据说这题是很基本的 AC 自动机上的 DP，但是网上的题解我看了好久才算自己弄明白，大概是因为我已经很久没有接触 DP 了，对 DP 没有什么感觉。因此下面我尽量写得详细一点。

首先是 fail 指针的建立。和一般的 AC 自动机 fail 指针建立的区别在于，以往广搜的时候如果 $p \rightarrow next[i] = NULL$ ，我们将不作处理，现在需要将 $p \rightarrow next[i]$ 指向 $p \rightarrow fail \rightarrow next[i]$ (p 非 root) 或者 root ($p = root$)。这么处理的原因在下面将解释。还有一点，找到失败指针指向的位置 pos 以后，需要

$p \rightarrow next[i] \rightarrow flag = pos \rightarrow next[i] \rightarrow flag$ 来更新 $p \rightarrow next[i]$ 是否为结尾。

然后再讲 DP。这题要用 DP 就一定需要用预分配的结点数组（手动管理内存池），因为需要枚举所有的 Trie 树节点。假设我们正在构造一个不含任何致病片段的 DNA 片段，设 $dp[i][j]$ 是已经构造完第 i 个碱基，并且第 i 个碱基在 Trie 树中是第 j 个结点，此时构造串对比起原串改变的碱基数。需要注意的是， i 的顺序是递增有意义的，也就是 $dp[i+1]$ 的结果要依赖于 $dp[i]$ 的结果；但是 j 的顺序是无意义的，我一开始一直没有想明白这一点。

一开始， dp 被初始化为 -1，意味着所有状态不可达，只有 $dp[0][0] = 0$ 。对于每一个 i ，我们枚举所有使得 $dp[i-1][j] \neq -1$ 的 j ，也就是假设第 $i-1$ 个碱基已经在第 j 个结点了。然后再枚举四种碱基，作为第 i 位的碱基。如果枚举的碱基，使用这个碱基后匹配时在树上将转移到第 k 个结点，而第 k 个节点是某个致病 DNA 片段的结尾，那么不使用。如果枚举的碱基和原来的目标 DNA 片段上相同位置的碱基相同，则 $dp[i][k] = \min(dp[i-1][j] + 1, dp[i-1][k])$ 。如果 $dp[i][k] = -1$ 的话就直接更新其值。

这么做的原因是，因为我们不关心改变的是哪几个碱基，只关心当前的目标 DNA 序列的后缀会不会和某个致病 DNA 片段相同，所以只需要保存当前匹配的结点编号就可以了。而 $p \rightarrow next[i] = NULL$ 时的处理是为了让匹配能够继续。至于更新 $flag$ ，是因为在 Trie 树上进行匹配的时候，我们不会跳转到失败指针去检查对于上一个碱基来说这一个碱基是否为致病 DNA 的结尾，只能在处理上一个碱基的时候就先进行判断。

P0J 1625 【中等】

题目大意：

给出一个含 N 个 ASCII $0 \sim 255$ 的字母表 ($1 \leq N \leq 50$)，求有多少个由字母表中给出的字符组成的长度为 M ($1 \leq M \leq 50$) 个字符的字符串，满足字符串中没有给定的 P 个 ($1 \leq P \leq 50$) 长度不超过 20 个字符的敏感词。

输入：

第一行有三个空格分隔的整数 N、M 和 P。
第二行有 N 个 ASCII 字符，表示新的字母表。
下面有 P 行，每一行有一个敏感词字符串。

输出：
一行，一个整数，即满足要求的字符串数量。

题解：
这题也是在 AC 自动机上的 DP。但是说是 DP，实际上更像是利用杨辉三角求组合数那种 DP，即分步计算。首先，由于字母表的范围很大，而 gcc 里面 char 的类型对应的整数是 -127~127，所以需要先把输入的字母表重新映射编号，也就是 $\text{Alphabet}[s[i]+128]=i$ 。建树和求 fail 指针和 POJ 3691 的完全一样（这两部分我的代码基本上丝毫未改）。至于 DP，也比较简单，用 POJ 3691 题解中的各字母含义， $dp[i][k]=dp[i][k]+dp[i-1][j]$ ，避开敏感词的结尾不断累加就可以了。由于这题结果肯定会很大，所以需要用到大数加法，我写了一个很简单的 BigInt 类。

HDOJ 2825 【中等偏上】

题目大意：
给出 M 个长度不超过 10 个字符的字符串（ $1 \leq M \leq 10$ ）（下面称之为关键串），要猜测的口令长度为 N（ $1 \leq N \leq 25$ ）个字符串，并且已经知道口令中一定含有至少 K 个给出的关键串。问要猜测多少次才能枚举完所有的可能。

输入：
有若干组测试数据。每一组测试数据第一行有三个整数 N、M 和 K， $N=M=K=0$ 时测试数据结束。接下来有 M 行，每一行有一个长度不超过 10 个字符的关键串。

输出：
每一组测试数据输出一行，包含一个整数，即所有可能的口令的数量模 20090717。

题解：
很显然这题也是在 AC 自动机上进行状态转移 DP 的题目，不过要进行状态压缩。我还没有系统地去练 DP，所以一开始对“状态压缩”这个叫法感到有点不解，现在看来应该是指增加一维用二进制表示某些选项选取哪些所形成的组合。那么，建 Trie 树的时候，每个节点的 flag 标志表示这个节点是哪一个关键串的结尾（ $\text{flag}=1 \ll \text{id}$ ）。设 $dp[i][j][k]$ 表示前 i 个字符，在 Trie 树上是第 j 个节点，并且出现了 k 对应的二进制状态中的几种关键串，并且第 j 个节

点的 $\text{next}[\text{pos}]$ 子节点在 Trie 树中是第 jj 个节点，出现了 kk 对应的二进制状态中的几个关键串，那么 $\text{dp}[\text{i}][\text{jj}][\text{k}|\text{kk}] += \text{dp}[\text{i}-1][\text{j}][\text{k}]$ 。注意使用的是或操作，因为前后两个节点会有共同包含的关键串。最后，将 $\text{dp}[\text{n}]$ 中所有满足关键串出现数量大于等于 K 的 $\text{dp}[\text{n}][\text{j}][\text{k}]$ 加起来就可以了。注意状态转移方程和最后的求和都要一边加一边模。还有，不知为何这题初始化 $\text{dp}[]$ 的时候 memset 会不如三重 for 循环来得快。

HDOJ 4057 【中等偏上】

题目大意：

给出 M 个 ($1 \leq \text{M} \leq 10$) 目标基因，每一个基因不超过 100 个碱基，有一个整数的价值。现在要求构造一个长度为 N 个碱基的序列，要求这个序列的价值最高。注意，如果一个目标基因在序列中出现多次，它的价值只计算一次。

输入：

有若干组测试数据。每一组测试数据第一行有两个整数 M 和 N ，接下来有 M 行，每一行有一个目标基因。

输出：

如果得到的最高价值的序列的价值小于 0，输出一行 “No Rabbit after 2012!”，否则输出一行包含一个整数，即最高的价值。

题解：

这题是我做完 HDOJ 2825 以后做的，所以几乎不费什么力气。事实上，代码也没有改动多少。还是状态压缩 DP。设 $\text{dp}[\text{i}][\text{j}][\text{k}] = 1$ 表示前 i 个字符，在 Trie 树上是第 j 个节点，并且出现了 k 对应的二进制状态中的几种目标基因，是可行的。取 jj 、 kk 同 HDOJ 2825 题解中的含义，如果 $\text{dp}[\text{i}-1][\text{j}][\text{k}] = 1$ ，则有 $\text{dp}[\text{i}][\text{jj}][\text{k}|\text{kk}] = 1$ ，连相加都不需要，因为我们不需要知道有多少个构造方法，只想知道有没有办法构造出来。最后将 $\text{dp}[\text{n}]$ 中所有大于 0 的 $\text{dp}[\text{n}][\text{j}][\text{k}]$ ，求其对应的 k 包含的目标基因的价值和，然后取最大值即可。

POJ 2778 【难】

题目大意：

有 M 种 ($1 \leq \text{M} \leq 10$) 致病 DNA 序列，每一种的长度不超过 10 个碱基。问有多少个长度为 N ($1 \leq \text{N} \leq 200000000000$) 个碱基的 DNA 序列，其中不包含任何给出的致病 DNA 序列。考虑到结果很大，模 100000 后输出。

输入：

第一行有两个整数 M 和 N，用空格分隔。下面有 M 行，每一行有一个致病 DNA 序列。

输出：

一行，一个整数，即总数模 100000。

题解：

这题是 Matrix67 菊苣的《十个利用矩阵乘法解决的经典题目》里的第 10 题。我觉得需要说明的一点是，需要使用静态的 Trie 树，也就是把指针全部转换为数字下标。这样，可以由 fail 指针得到一个状态转移矩阵。此前的 DP 是相当于直接对这个矩阵做多次状态转移。现在由于次数特别多，所以可以使用矩阵相乘的快速幂来解决。只要统计一下最后得到的矩阵里第一行的和就可以了。

HDOJ 2243 【难】

题目大意、输入输出要求见原题中文。

题解：

这题是 POJ 2778 的加强版。因为有词根出现的单词长度不定，从 1 到 n 都可以，因此按 POJ 2778 里的静态 Trie 树转矩阵后做快速幂，要做 $A + A^2 + \dots + A^n$ ，而所有单词的数量是 $26 + 26^2 + \dots + 26^n$ ，因此最后的结果是 $26 + 26^2 + \dots + 26^n - (A + A^2 + \dots + A^n)$ 。 $A + A^2 + \dots + A^n$ 的求法在 POJ 3233 里我已经写过了， $26 + 26^2 + \dots + 26^n$ 也可以用矩阵来求。

这题有一个比较坑的地方是，模 2^{64} 。实际上，unsigned __int64 最大的范围就是 $2^{64}-1$ ，而超过这个范围则从 0 开始重新计数，相当于自动模余了。这样就不需要在矩阵乘法的时候一边乘一边模余了。

HDOJ 3341 【难】

题目大意：

给出 N 个 ($1 \leq N \leq 50$) “数论基因”，一个 DNA 序列中每一个“数论基因”可以增加 1 点的数论能力，每个“数论基因”的长度不超过 10 个碱基。然后给出一个 DNA 序列，可以重组这个 DNA 序列中的碱基顺序，问最大能获得多少点的数论能力。

输入：

有若干组测试数据。每一组测试数据第一行有一个整数 N ， $N=0$ 时测试数据结束。接下来有 N 行，每一行有一个“数论基因”。最后有一行，包含一个 DNA 序列，不超过 40 个碱基。

输出：

每一组测试数据输出一行，格式为 Case %测试数据组编号%: %最大数论能力%。其中测试数据组编号从 1 开始。

题解：

做了几题状态压缩 DP，一开始我没有想到这个要怎么压，只是想到了最直接的开 5 个维度来进行 DP，但是显然时间和空间上都不允许这么做。查了一下，网上的题解很多说得并不清楚，我看了几篇所谓的变进制状态压缩以后自己摸索了很久，总算搞明白了。下面我写一个详细一点的变进制状态压缩 DP 原理。首先，每一个 Trie 树结点上的 flag 是用来维护这个节点是多少个“数论基因”的结尾的，因为“数论基因”出现多次按多次计算。这是我一开始没有搞清楚的地方，也是最后改过来的一个错误。因此，在构建 fail 指针的时候，需要 $p \rightarrow \text{next}[i] \rightarrow \text{flag} += \text{pos} \rightarrow \text{next}[i] \rightarrow \text{flag}$ ，其中 pos 是 p 的失败指针指向的结点，原理同 POJ 3691 题解中所说的。

然后，我们需要统计被更改的基因串中各碱基的数量 $\text{num}[i]$ 。设 ACTG 对应 0~3，建立 base 数组， $\text{base}[i]$ 表示第一个序号为 i 的碱基的出现对应的状态编号。 $\text{base}[3]=1$ ， $i=2 \sim 0$ 时 $\text{base}[i] = \text{base}[i+1] * (\text{num}[i+1] + 1)$ ，这样就保证了所有的碱基组合都可以得到编号。那么枚举每一种碱基组合编号的时候，只要对应地除和模余就可以得到该编号对应的各个碱基的数量。

设 $\text{dp}[i][j]$ 表示采用编号为 i 的碱基组合时转移到 Trie 数上的第 j 个结点，所能得到的数论能力。如果增加一个碱基 k ，那么新的碱基组合编号为

$i + \text{base}[k]$ ，转移到的点的编号为 jj ，很显然

$\text{dp}[i + \text{base}[k]][jj] = \text{dp}[i][j] + \text{mem}[jj].\text{flag}$ ，mem 是存储各 Trie 结点的数组。

在转移的时候，只要新增的碱基数不会超过给定的碱基数，那么就可以执行转移。

最后，统计一下最后一层的 dp 中的最大值，就得到了结果。