**G - DNA序列**

**Time Limit: 1000/300MS (Java/Others)     Memory Limit: 65535/65535KB (Java/Others)**

Submit Status

众所周知，DNA序列是仅含有A，C，TA，C，T和GG的序列，分析DNA序列的一段非常有用，例如，如果动物的DNA序列含有片段ATCATC，则可能意味着可能有 一种遗传性疾病。

到目前为止，科学家已经发现了几个片段，问题是一个物种的DNA序列有多少种不包含这些片段。

假设物种的DNA序列是由A，C，TA，C，T和GG组成的序列，序列长度为给定的整数nn。

**Input**

第一行包含两个整数m(0≤m≤10)m(0≤m≤10)，n(1≤n≤2000000000)n(1≤n≤2000000000)。 这里，mm是遗传疾病段的数量，nn是序列的长度。

接下来mm行每行表示一个DNA遗传疾病段，并且这些段长度不为零且不大于1010。

**Output**

一个整数，DNA序列数，modmod 100000100000。

**Sample input and output**

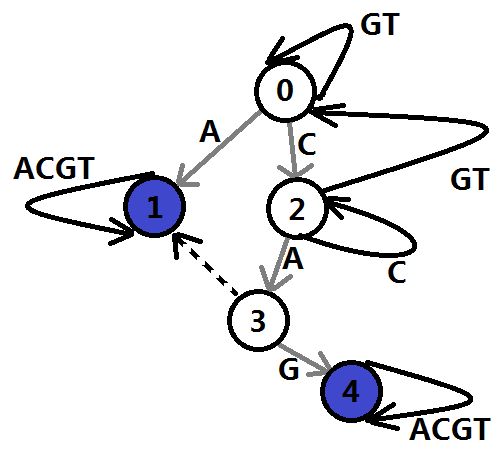
| **Sample Input** | **Sample Output** |
| --- | --- |
| 4 3  AT  AC  AG  AA | 36 |

题解：

较为复杂。

首先，根据给定的字符串构建AC自动机。观察可得，模式串长度很小，所以可以利用数组模拟链表。对于每个节点，增加一个标记，表示如果来到该点是否可以继续构建字符串。

下图：模式串A和CAG构建的自动机，加上一些没有提到的字符，则构建了trie图。蓝点为不可达的点。



构建fail的时候要注意，如果fail不可达，则其自身也不可达。

这样问题可以变为，从树根走，怎么走可以避免走进终止状态。我们将每个点看做一个dp元素，dp[i][j]为前i个字符在j点的方案数，则问题可以转换为动态规划问题。Dp[i][j]只与dp[i-1]有关，所以可以利用自动机中的边构造邻接矩阵，两点有边，则代表边指向的点多了一种转化方法。转化次数为m次，利用快速幂求解。答案就是dp后根节点的方案数。

不得不说，实现有一定难度。