**10298: Power Strings**

★★☆☆☆

* **題組：**Problem Set Archive with Online Judge
* **題號：**10298: Power Strings
* **解題者：陳衍豪**
* **解題日期：**2014年3月20日
* **題意：**

定義a\*b為a和b兩個字串的連接式。舉例來說：

 若a = “abc”，b = “def”，則a\*b 定義為“abcdef“。

 今天程式要求當使用者輸入包含可印出字元的字串s時，

 程式必須判斷使s = a ^ n的n值，也就是s最長可由幾個子字串組成。其中s的範圍介於一個字元至最長10 ^ 6個字元。

 輸入由 ” . ” 字元代表結束。

* **題意範例：**

input：　　output：

 　 abcd　　　 1

 　 aaaa　　　 4

 　 ababab 　 3

* **解法：**

**(1)窮舉法:**

0. 1. 2.

ababcabc ababcabc ababcabc

||| ||| |||

abc abc abc

(X) (X) (O)

3. 4. 5.

ababcabc ababcabc ababcabc

 ||| ||| |||

 abc abc abc

 (X) (X) (O)

時間複雜度 = O(a字串長度\*b字串長度)

* 會超過題目所規定的時間限制

**(2) KMP(Knuth-Morris-Pratt)演算法:**

＊次長的共同前後綴(Longest Proper Prefix-Suffix)：

**T: aabzabzabcz P: abzabc**

 當挪動至下圖位置，發現P僅有一部分比對成功：

 V

 aabzabzabcz

 |||||

 abzabc

.abzab..... .abzab..... .abzab..... .abzab..... .abzab.....

 |||| ||| || |

 abzab. abzab. abzab. abzab. abzab.

 (P shift 1) (P shift 2) (P shift 3) (P shift 4) (P shift 5)

得到比對成功的字串片段，也就是P的前綴(prefix)為abzab 它的「次長的共同前後綴」：ab。

如果預先知道「次長的共同前後綴」，就可以一口氣大幅挪動P，略過許多步驟：

 V V

 aabzabzabcz aabzabzabcz

 ||||| ---> ||

 abzabc abzabc

由「V」處繼續向右比對字元。當比對失敗、遇到相異字元，就再次使用比對成功的字串片段，取其「次長的共同前後綴」來大幅挪動P。

**＊失敗函數：failure function （ prefix function ）：**

它是一個字串函數，輸入字串的其中一個前綴，則輸出該前綴的「次長的共同前後綴」。

 位置 0 1 2 3 4 5 6

 T[7] = A A B A A B B

 F[7] = -1 0 -1 0 1 2 -1

以下為程式碼

for (int i=1, j=failure[0]=-1; i<p.size(); ++i)

{

　　while (j >= 0 && p[j+1] != p[i])

 j = failure[j];

　　if (p[j+1] == p[i]) j++;

　　failure[i] = j;

}

* **解法：**

在主程式中由一個參數t去接收KMP函式所回傳的j值，並判斷字串長度len是否整除(len-t-1)，是則印出其商數，否則印出1。

* **解法範例：**

abcd　 len=4　 F[4] = -1 -1 -1 -1 return j = -1

aaaa　 len=4　 F[4] = -1 0 1 2 return j = 2

ababab len=6 F[6] = -1 -1 0 1 2 3 return j = 3

* **討論：**

對於 字串s的某一個字元來說，與其他字元進行比對的次數**<=**「當下比對成功的字串片段」的長度。「當下比對成功的字串片段」是動態改變的，所以字元兩兩比對的總次數不超過 2\*s 次。同理計算失敗函數的總次數不會超過2\* 「次長的共同前後綴」次，又最長的「次長的共同前後綴」不會超過s。

* **時間複雜度 = O(s)**