

# 문자열찾기

**[문제]** 스트링 매칭이란 주어진 텍스트  $T$ 에서 찾고자 하는 패턴  $P$ 를 찾는 문제이다. 문자열 패턴을 찾는 알고리즘이 얼마나 차이가 나는지 알아 보자. 특히 비슷해 보이는 Rabin-Karp(RK), KMP, Boyer Moore(BM) 알고리즘들의 패턴을 찾기위한 비교횟수를 관찰해 보자. 비교횟수가 적은 순서대로 RK, KMP, BM라는 알고리즘 이름을 출력한다. 예를들어, 텍스트 가 FAFEDFDBEF가 주어지고 패턴 는 FDBEF이다. 각 방법을 사용할 때, 비교횟수는 RK는 11회, KMP는 12회, BM는 7회이다. 따라서 BM이 가장 빠르고 RK, KMP 순이다. 출력은 BM RK KMP 이다.

RK방법은 오버플로우를 방지하기 위해 모듈연산을 하는데 이 때 사용하는 방법은 다음과 같다. 어떤  $i$  위치에서의  $m$  크기의 부분문자열의 값을

$$b_i = (d(b_{i-1} - (d^{m-1} \bmod q)A[i-1]) + A[i+m-1]) \bmod q$$

을 사용한다. 이 때,  $A[i]$ 는 위치  $i$ 에서 문자의 값이다. 그리고  $m$ 개의 문자열의 수치값이 같을 때는 실제 문자가 같은지 검사해야 한다. 따라서 RK의 비교횟수는 수치값비교횟수와 문자비교횟수를 더한 것이다.

**[입출력]** 입력의 첫 줄에는 찾고자 하는 패턴  $P$ 의 줄 수와 다음 줄에는 패턴이 주어진 수 만큼 주어진다. 그리고 그 다음 줄부터 텍스트  $T$ 의 줄 수와 실제 텍스트가 주어진다. 패턴과 텍스트는 한 줄에 80자가 넘으면 다음 줄에 표시한다. 따라서 실제로 비교를 할 때는 80자 이상인 것은 모두 리턴값을 없애고 진행해야 한다.

출력은 패턴을 찾기 위해 실행된 비교횟수가 적은 순서대로 알고리즘 이름을 출력한다. Rabin-Karp을 RK, KMP알고리즘은 KMP, Boyer Moore알고리즘은 BM으로 출력한다. 아래 예에서 두번째 예제는 앞에서 설명한 것처럼 (RK, KMP, BM의) 각 알고리즘의 비교수가 BM, RK, KMP 순이다. 순서가 동일할 경우에는 0를 출력한다. 예를 들어 세 알고리즘의 비교횟수가 동일하면 0 0 0을 출력하고 KMP 비교횟수가 제일 적고 나머지 알고리즘 비교횟수가 같다면 KMP 0 0을 출력한다.

**[예제]**

입력 stdin	출력 stdout
1 //패턴의 줄 수 GAEAE //패턴 P 1 //텍스트의 줄 수 EEGAEAEDDC //텍스트 T	KMP 0 0

1 CBEBE 1 CBEBECFEDF	0 0 RK
1 GCCGC 1 EGFEAGCCGC	BM 0 0
1 CDBFA 1 DECDBFAEBD	BM KMP RK
1 ACABD 1 GAFAACABDF	BM RK KMP

**[제한조건]** 프로그램의 이름은 pa08\_matching.{py,c,cpp,java}이다. 제출 횟수는 최대 15번이며  
허용 시간은 데이터 당 제한 시간은 1초, 허용가능 코드의 최대 크기는 5,000 bytes 이다.  
문제 풀이 마감시간은 2024년 5월18일 24:00 이다.