

1. kmp.txt 에서 문자열을 읽어온다.
2. 패턴 문자열의 실패함수를 생성한다.
3. 실패함수란, 패턴 문자열의 특정 위치 i 로부터 앞으로 연속으로 읽을 때와 같은 수만큼 0번째 위치부터 뒤로 읽는다고 가정했을 때, 그 두 연속적인 문자열이 최대한 길게 같아질 수 있는 범위를 failure[i]에 저장하는 함수이다.
4. i가 증가함에 따라 앞의 문자들과 이어지는 경우에 실패함수값이 하나씩 올라가다 더 이상 이어지지 않는 지점이 발생하면, 다시 0부터 시작하는 것이 아니라 i의 바로 앞 지점과 같은 문자로 끝나는 처음부터 시작하는 연속적인 패턴을 다시 찾기 위해 대응되는 지점으로 가서 다시 비교해주는 것이 필요하다. 따라서

while ((pat[j] != pat[i]) && (i > 0)) {

i = failure[i - 1];

라는 구절을 넣어준다.

1. 실패함수를 바탕으로 매치함수를 실행한다.
2. 매치함수는 string 탐색의 시작지점 s와 s를 기준으로 한 위치 i가 필요하다. 전반적으로 실패함수와 비슷한 느낌을 가지며, 같은 pat 문자열과 하나씩 비교하는 과정을 거친다. 만약 처음부터 불일치하면 그대로 s만 1만큼 증가시키며 문자 일치가 나타나면 더 이상 같지 않을 때까지 i를 증가시키며 비교한다. 이후 i가 pat의 문자열 길이 이상이 된다면 패턴이 일치했다는 것이므로 pat과 같아지기 시작하는 string의 위치 s를 반환한다. 만약 중간에 불일치가 나타나면 i-1의 실패함수값을 확인한다. 실패함수값이 0이라면 그 지점은 패턴의 앞부분과 연속적으로 대응되는 것이 아무것도 없다는 의미이므로 그 지점까지는 더 이상 볼 필요가 없기 때문에 점프할 수 있다. 1이라면 실패함수 제일 앞 부분과만 연속적으로 대응된다는 의미이며 2이면 두 개까지 대응된다는 의미이다. 이런 식으로 일반화를 시키면, s = s + i - failure[i - 1] 이라고 할 수 있다. 이 때 i는 실패함수 값으로 지정해주면 일치하는 부분을 건너뛰고 이어서 패턴 일치여부를 확인할 수 있다. 만약 s가 범위 이상으로 증가하게 되면 패턴을 찾지 못 했다는 의미이므로 -1을 반환한다.