KMP

KMP 알고리즘은 건너뛰기 표를 만드는 방법이다.

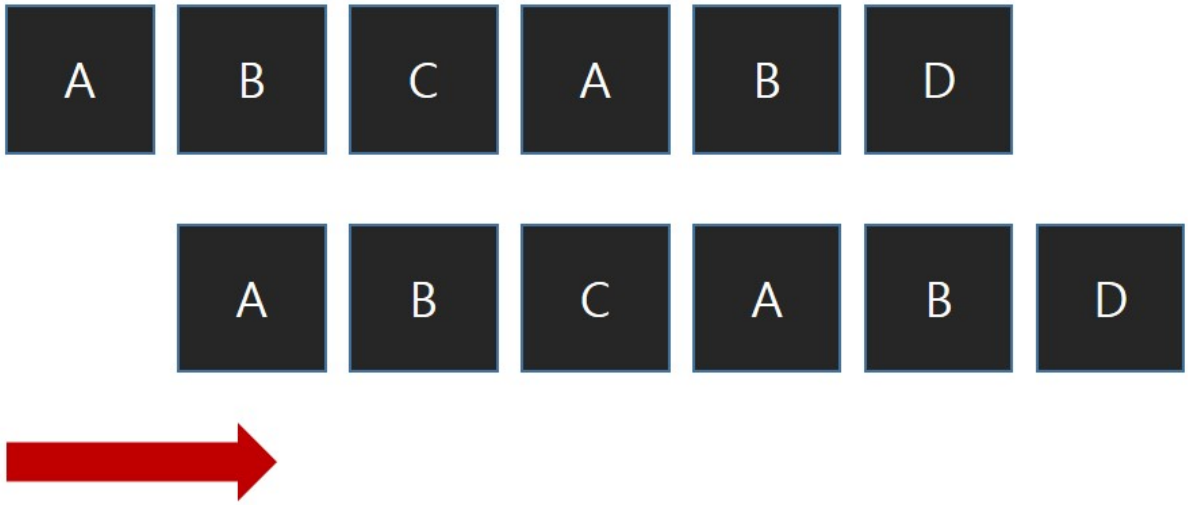
코드

```
def kmp_match(txt, pat):
    # pt는 다음 칸부터 봐야한다.
    pt = 1
    pp = 0
    # pattern의 크기만큼 skip표를 만든다.
    skip = [0] * (len(pat) + 1)

    skip[pt] = 0
    # pat의 크기만큼 본다.
    while pt != len(pat):
        # 만약에 같으면
        if pat[pt] == pat[pp]:
            # pt와 pp를 한칸 옮기고
            pt += 1
            pp += 1
            # 그 자리를 저장한다. pat의 n번째 자리가 겹쳐 있다는 뜻임.
            skip[pt] = pp
        # pp가 0이라면 pat에 겹치지 않다는 뜻
        elif pp == 0:
            pt += 1
            skip[pt] = pp
        else:
            pp = skip[pp]

    pt = pp = 0
    while pt != len(txt) and pp != len(pat):
        if txt[pt] == pat[pp]:
            pt += 1
            pp += 1
        elif pp == 0:
            pt += 1
        else:
            pp = skip[pp]

    return pt - pp if pp == len(pat) else -1

txt = 'ZABCABXACCADEF'
pat = 'CCAD'
```

```
print(kmp_match(txt, pat))
```

순서

1. skip표 만들기

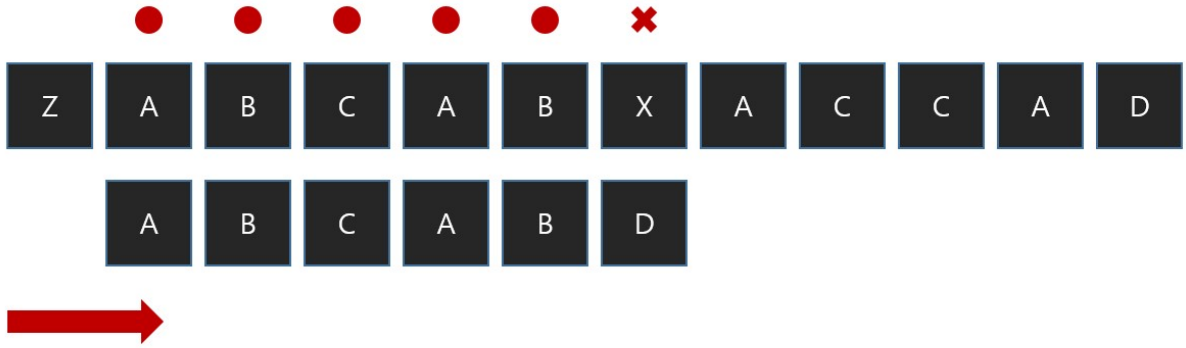


반복되는 문자열의 개수를 찾는다.
A부터 시작하여 2번 반복됨

A	B	C	A	B	D
-	0	0	1	2	0

2. 검색(건너뛰기가 가능한 경우)

다음 검색 자리인 C부터 보면된다.



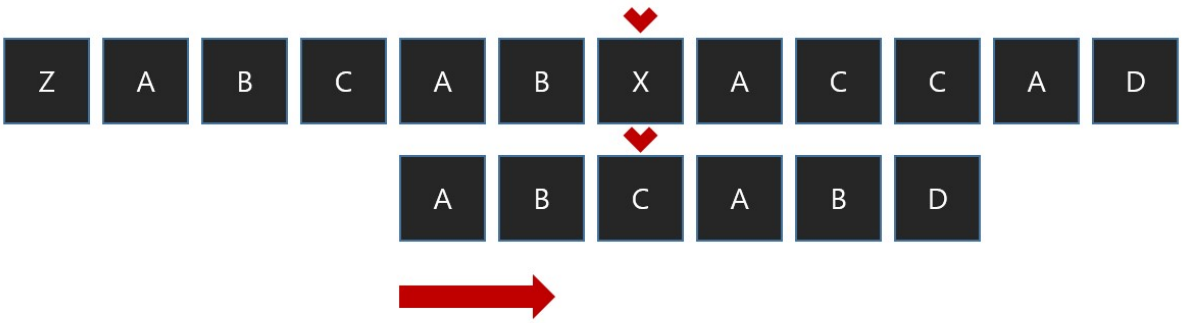
pp = 4
(틀리기 직전 시점)

pp = skip[pp]
(skip 표에서 찾기)
∴ pp = 2

pattern[2] = C 부터 본다
pt => 검색 틀린 자리

3. 검색(건너뛰기가 불가능한 경우)

처음부터 봐야한다.



pp = 1
(틀리기 직전 시점)

pp = skip[pp]
(skip 표에서 찾기)
∴ pp = 0

pattern[0] = A 부터 본다
pt => 검색 틀린 자리