

# Boyer-Moor

보이어-무어 또한 KMP와 마찬가지로 건너뛰기 표를 만들어 하는데

## 코드

```
def bm_match(txt, pat):

    # bm의 skip표는 가능한 많은 문자열
    # (한글이나 일본어나 중국어같은 것들이 더해진다면 바뀔 수 있음)
    skip = [None] * 256

    # skip 표 만들기
    for pt in range(256):
        # 모두 pat의 길이만큼 밀려날 가능성을 초기화 CCAD일 경우 4로 초기화함
        skip[pt] = len(pat)
        # 패턴을 돌면서 그 문자열 자리에 (pat의 전체 크기) - (문자열 위치) - 1을 해줌
        # 만약에 skip표의 값이 변하지 않았다는 것은 pat에 그 알파벳이 없다는 것.
    for pt in range(len(pat)):
        skip[ord(pat[pt])] = len(pat) - pt - 1

    while pt < len(txt):
        pp = len(pat) - 1
        while txt[pt] == pat[pp]:
            if pp == 0:
                return pt
            pt -= 1
            pp -= 1

        # 만약 현재 보던 txt의 문자열(검색 실패한 경우)의 skip 표의 값이
        # (패턴의 크기) - (pp의 위치) 보다 크다면 맞는게 없다는 뜻임으로 skip표에 적은만큼 쪽 이동한다.
        if skip[ord(txt[pt])] > len(pat) - pp:
            pt += skip[ord(txt[pt])]
        # 맞는게 있었다면 그만큼만 이동해줌.
        else:
            pt += len(pat) - pp

    return -1

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

```
print(bm_match(txt, pat))
```

# 순서

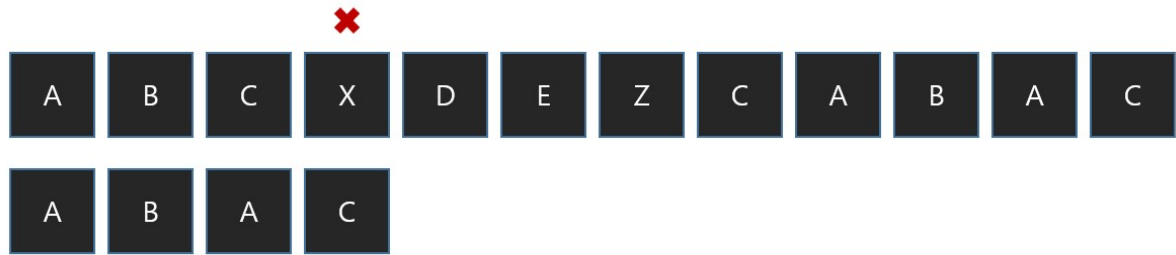
## 1. skip 표 만들기

몇 칸을 미뤄도 되는가 ?

A	B	C	D	E	F	G	H	I	J	K	L	M
1	2	4	4	4	4	4	4	4	4	4	4	4
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
4	4	4	4	4	4	4	4	4	4	4	4	4

패턴 : 'ABAC'  
A가 나올 경우 3번째 것인지  
1번째 것인지 알지 못하니 최대 1칸 미룰 수 있음

## 2-1. skip 표에 맞춰서 이동하기(맞는 게 하나도 없는 경우)

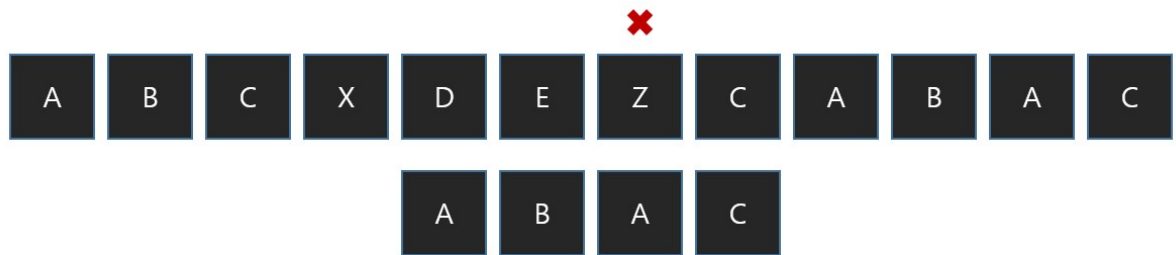


X가 끝에서 틀렸음으로  
4칸 이동

if skip[ord(txt[pt])] > len(pat) - pp  
4 > 4-3

∴ pt += skip[ord(txt[pt])]  
( pt += 4 )

## 2-2. skip 표에 맞춰서 이동하기(맞는 게 하나라도 있는 경우)



Z가 끝에서 틀렸음으로 4칸 이동  
(Z에서 4칸 이동)

if skip[ord(txt[pt])] > len(pat) - pp  
4 > 4-2

∴ pt += skip[ord(txt[pt])]  
( pt - 1 += 4 )