

KMP

일반적인 방법 : $O(|A||B|)$
KMP : $O(|A| + |B|)$

용어 정리

- $S[a:b] = "S[a] S[a+1] \dots S[b-1]"$
- $S = "ABCDEF"$ 일 때 $S[2:5] = "CDE"$, $S[3:4] = "D"$, $S[0:2] = "AB"$
- $|S|$ 는 S 의 길이, $S = "ABCDEF"$ 일 때 $|S| = 6$
- 접두사 = 문자열의 첫 문자를 포함하는 연속한 문자열, $S[0:x]$
- 접미사 = 문자열의 끝 문자를 포함하는 연속한 문자열, $S[x:|S|]$
- A, AB, ABC, ABCD, ABCDE, ABCDEF는 ABCDEF의 접두사
- F, EF, DEF, CDEF, BCDEF, ABCDEF는 ABCDEF의 접미사

- KMP : 패턴 매칭 문제를 $O(|A| + |B|)$ 에 해결할 수 있는 기적의 알고리즘
- 쉬움
- 먼저 KMP에서 쓰이는 **실패 함수**를 알면 KMP를 이해하는데 도움이 됨

failure function

```
vector<int> failure(string& s) { //failure function
    vector<int> f(s.size(), 0);
    int j = 0;

    for (int i = 1; i < s.size(); i++) {
        while (j > 0 && s[i] != s[j]) {
            j = f[j - 1];
        }

        if (s[i] == s[j]) {
            f[i] = ++j;
        }
    }

    return f;
}
```

ABABCABABA

f 0 0 1 2 0 1 2 3 4 3

단순히 A 문자열 안에 B 문자열이
등장하는지 알고 싶으면 strstr 함수 쓰기

KMP는 다양한 응용이 가능하다

B 문자열을 찾은 후 탐색을 더 해야 한다면
 $j = F[j-1]$ 하면 된다