편집 거리

편집 거리(Edit Distance = Levenshtein Distance)란 두 문서[문자열] 간의 유사도를 판단하는 데에 사용하는 알고리즘이다. 여기서 **거리**란, 형태적인 거리일 뿐이고, 의미적 거리가 아니다.

> 예컨대, 문서 편집 프로그램에서 단어를 타이핑하다 오타를 낸 경우를 생각해 보자. 오타가 난 단어에 빨간 줄이 그이고, 추천 단어가 나온다. 이 때 추천 단어를 보여줄 때 편집 거리가 작은 순서대로 추천된다.

1. 알고리즘

한 문자열을 다른 문자열로 만들기 위해 다음과 같은 **편집 연산**을 사용할 수 있다.

- 삭제
- 삽입
- 대체(치환)

한 문자열을 다른 문자열로 만들기 위해, 문자열의 철자를 하나씩 비교하며 얼마나 많은 편집 연산이 필요한지 판단한다. 최소한의 편집 연산의 수를 편집 `cost`로 본다.

교재 90페이지에 있는 예시를 통해 편집 cost를 계산하자. 행과 열에 바꿀 문자열을 놓는다.

1. 처음 비교 대상은 공집합(Φ)이다.
    - 공집합과 공집합은 같은 문자열이기 때문에 바꿀 것이 없다.
    - 공집합에서 다음 문자열로 가기 위해 하나씩 추가해야 하기 때문에, 1행과 1열의 cost는 0에서 1, 2, 3과 같이 1씩 증가한다.

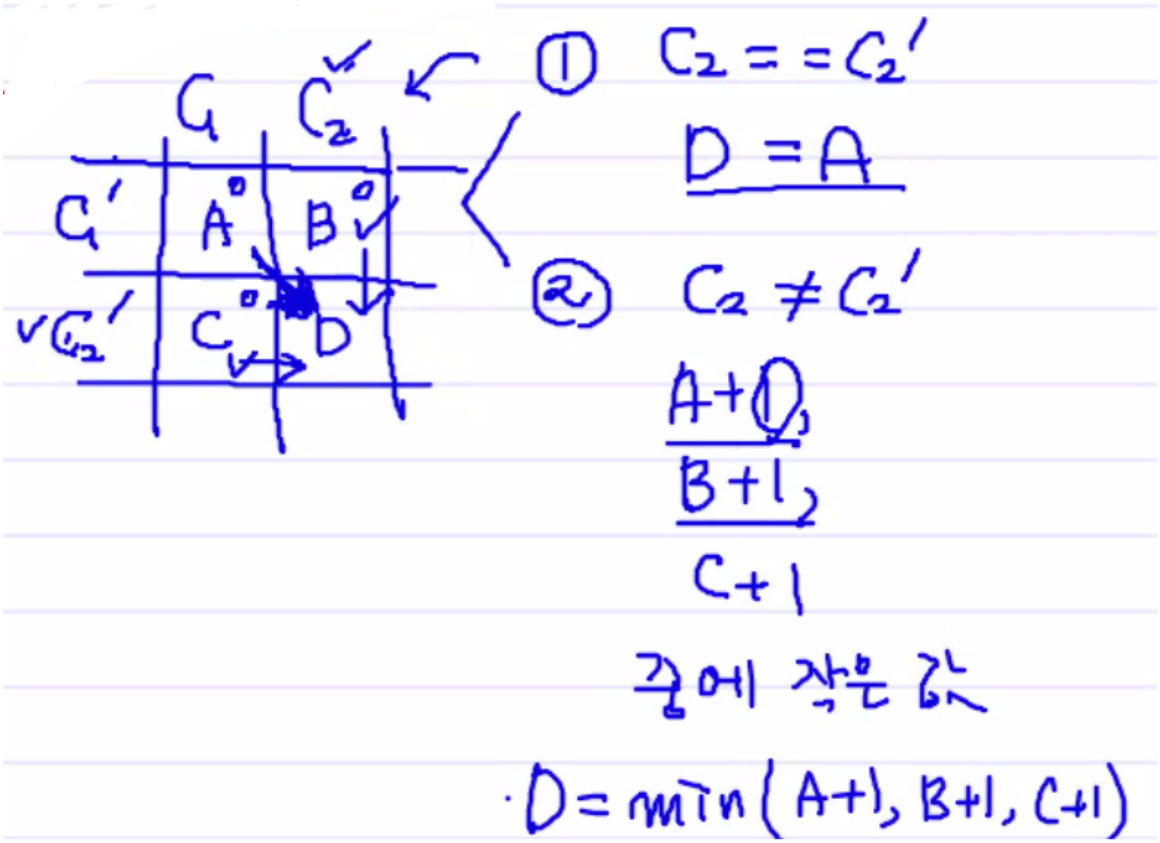
	Φ	**H**	**A**	**N**	**D**
Φ	0	1	2	3	4
A	1				
N	2				
D	3				

2. `(m, n)` 열의 `cost`를 계산하기 위해, 이전까지의 문자열와 누적하여 비교한다. 다음의 예시 몇 가지를 살펴 본다.
    - `(1, 1)` : `A`를 `H`로 만들기 위해 대체한다.
    - `(2, 2)` : `AN`을 `HA`로 만들기 위해 `A`를 `H`로 대체하고, `N`을 `A`로 대체한다. 2번의 연산이 필요하다.
    - `(2, 3)` : `AN`을 `HAN`으로 만들기 위해 `H`를 추가한다. 1번의 연산이 필요하다.
    - `(3, 3)` : `AND`를 `HAND`로 만들기 위해 `H`를 추가한다. 1번의 연산이 필요하다.

	Φ	**H**	**A**	**N**	**D**
Φ	0	1	2	3	4
A	1	1	1	2	3
N	2	2	2	1	2
D	3	3	3	2	1

이렇게 채운 표의 맨 마지막 요소가 한 문자열을 다른 문자열로 바꾸기 위해 필요한 연산의 최소 수가 된다.

위의 알고리즘을 코드로 구현하기 위해 규칙을 찾으면 다음과 같다.



C_2와 C_2'가 만나는 칸이 최소 연산 수가 된다.

- 만약 두 개가 같다면 아무런 편집 연산도 필요하지 않으므로, A = D가 된다.
- 그렇지 않다면, 미리 구해 놓은 A, B, C 중 최솟값에 1번의 편집 연산만 더 하면 된다.

> DP원리를 따르는 듯(?)하다.

2. 코드 구현

실제로 이를 로직으로 구현하고, `nltk` 라이브러리에서 import해서 사용한 뒤, 두 차이를 비교해 보자.

```
from nltk.metrics.distance import edit_distance

def my_edit_distance(str1, str2):
    m = len(str1) + 1
    n = len(str2) + 2

    table = {}
    for i in range(m):
        table[i, 0] = i
    for j in range(n):
        table[0, j] = j

    for i in range(1, m):
        for j in range(1, n):
            cost = 0 if str1[i-1] == str2[j-1] else 1 # 같으면 0(더할 연산이 없음), 1(한 번 연산 더 해야 함.)
            table[i, j] = min(table[i, j-1] + 1,
                              table[i-1, j] + 1,
                              table[i-1, j-1] + cost)

    return table[i, j]

print(my_edit_distance('HAND', 'AND'))
print(edit_distance('HAND', 'AND'))
```