

# 알고리즘

과제번호	12주차
날 짜	2018.11.29
학 번	201302395
이 름	류경빈

## 과제 1

## Sequence Alignment

$$p(i, j) = \begin{cases} +1, & \text{if } x[i] = y[j] \\ -1, & \text{if } x[i] \neq y[j] \end{cases}$$

두 입력 단어의 동일성을 알아내는 배열

```
// p 배열 만들기
public void makeP(int lengthX, int lengthY){
    for (int i = 1; i < lengthY; i++) {
        for (int j = 1; j < lengthX; j++) {
            if (x[j].equals(y[i])){
                p[i][j] = 1;
            }else {
                p[i][j] = -1;
            }
        }
    }
}
```

p배열 알고리즘 구현

$$a[i, j] = \max \begin{cases} a[i, j-1] - 2 \\ a[i-1, j-1] + p(i, j) \\ a[i-1, j] - 2 \end{cases}$$

```
// 가로 세로 첫 번째 줄 배열에 초기 값 입력
for (int i = 0; i < lx; i++){
    a[0][i] = i*q;
}
for (int j = 0; j < ly; j++){
    a[j][0] = j*q;
}
```

```
for (int i = 1; i < ly; i++){
    for (int j = 1; j < lx; j++){
        int maxValue =
            MAX( a[i][j-1] + q,
                a[i-1][j-1] + p[i][j],
                a[i-1][j] + q );
        a[i][j] = maxValue;
    }
}
```

a배열 알고리즘 구현

## 결과 화면

17번째 과정

0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
-2	1	-1	-3	-5	-7	-9	-11	-13	-15	-17
-4	-1	2	0	-2	-4	-6	-8	-	-	-
-6	-	-	-	-	-	-	-	-	-	-
-8	-	-	-	-	-	-	-	-	-	-
-10	-	-	-	-	-	-	-	-	-	-
-12	-	-	-	-	-	-	-	-	-	-
-14	-	-	-	-	-	-	-	-	-	-
-16	-	-	-	-	-	-	-	-	-	-
-18	-	-	-	-	-	-	-	-	-	-

중간 과정

90번째 과정

0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20
-2	1	-1	-3	-5	-7	-9	-11	-13	-15	-17
-4	-1	2	0	-2	-4	-6	-8	-10	-12	-14
-6	-3	0	1	-1	-1	-3	-5	-7	-9	-11
-8	-5	-2	-1	2	0	0	-2	-4	-6	-8
-10	-7	-4	-3	0	3	1	-1	-3	-3	-5
-12	-9	-6	-5	-2	1	2	0	-2	-2	-2
-14	-11	-8	-7	-4	-1	2	1	-1	-3	-3
-16	-13	-10	-7	-6	-3	0	3	2	0	-2
-18	-15	-12	-9	-8	-5	-2	1	2	3	1

첫 번째 입력 값 : actgagttaa  
두 번째 입력 값 : acagaagta-  
점수 : 1

첫 번째 입력 값 : actgagttaa  
두 번째 입력 값 : acagaagt-a  
점수 : 1

최종 결과

해당 조합을 나타내기 위해 두 개의 입력 값의 크기+1만큼의 이차원 배열을 생성해 해당 입력 값을 비교하고 점수를 매겨 배열을 완성한다. 해당 배열은 그래프로 XY 그래프로 생각해서 왼쪽, 대각선, 위쪽 세 가지의 경우를 가지고 경로를 정한다. (lx, ly)에서 (0,0)까지 가는 경로를 통해 1:1 매칭을 구할 수 있다.