알고리즘

- HW08: Sequence Alignment-

제 출 일	2017년 11월 22일
분 반	02반
담당교수	공은배
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. 해결 방법

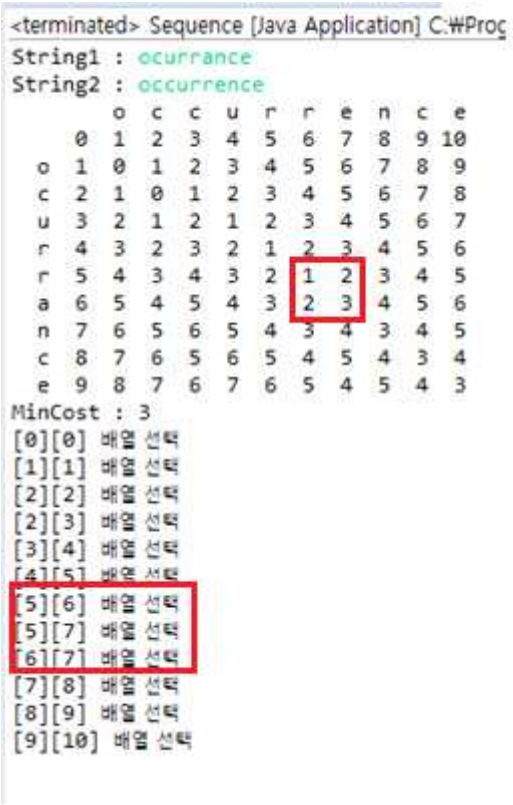
```
int[][] diff = new int[str1.length() + 1][str2.length() + 1];
// 각자의 길이만큼의 2차원 배열 선언
for (int i = 0; i <= str1.length(); i++) {
    diff[i][0] = i;
}
for (int i = 0; i <= str2.length(); i++) {
    diff[0][i] = i;
} // 0번째 즉 모든 수치를 갭으로 계산하여서 채워준다.
for (int i = 1; i < str1.length() + 1; i++) {
    for (int j = 1; j < str2.length() + 1; j++) {
        if (str1.charAt(i - 1) == str2.charAt(j - 1)) {
            diff[i][j] = diff[i - 1][j - 1];// 만약 문자열이 같다면 대각선왼쪽 윗값을 선택한다.
        } else {
            int A = 1 + diff[i - 1][j];
            int B = 1 + diff[i][j - 1];
            int C = 2 + diff[i - 1][j - 1];
            int diffmin = Math.min(A, B);
            diff[i][j] = Math.min(diffmin, C);// A,B,C중에 가장 작은값을 저장한다.
        }
    }
}
```

이전과제와 같이 OPT테이블을 만드는건 쉬웠다.
I,J의 값을 선택할 때 같은 문자열이면 I-1,J-1값을 그대로 저장하고 아닐경우에는 I,J의 주변값과 GAP을 할지 missMatch를 할지 선택해야한다 gap의 비용은 1이고 missMatch의 비용은 2이다.
여기서 이 코드의 문제점을 발견했다 바로 gap을 2번하는거나 missMatch 1번의 비용이 같아서 최소값을 구할 때 missMatch가 전혀 실행 안 되도 된다.
물론 최소를 구하는점에서는 어떠한 점을 이용하여도 문제가 없다.
과제 예시에도 missMatch로 가도 되는 경우가있는데 전부다 Gap으로 가고 있다.
이 문제로 나오는 오류는 추후 최단루트 찾을 때 다루도록 하겠다.

```
// 이제 어떤 루트로 생성되었는지 알아보는 코드이다
int col = str1.length();
int row = str2.length();
//역으로 추적하는게 구현으로 더쉽기때문에 시작점을 끝점으로 하였다.
while (col != 0 || row != 0) {
    int minI, minJ;
    if (col == 0) {
        minI = col;
        minJ = row - 1;
        row--;
    } else if (row == 0) {
        minI = col - 1;
        minJ = row;
        col--;//만약 행이나 열이 0이라면 바로 감소시키고
    } else if ( diff[col - 1][row] > diff[col - 1][row - 1]
            && diff[col][row - 1] > diff[col - 1][row - 1]) {
        minI = col - 1;
        minJ = row - 1;
        col--;
        row--;
    //현재 행과 열에서 왼쪽,위쪽,대각선 왼쪽 위중 가장 작은 값을 인덱스를 구하는 과정이다.
    }
     else if (diff[col - 1][row] > diff[col][row - 1] && diff[col - 1][row - 1] > diff[col][row - 1]) {
        minI = col;
        minJ = row - 1;
        row--;
    } else {
        minI = col - 1;
        minJ = row;
        col--;
    }
    MinRoute.add(new node(minI, minJ));
    //여기서 나온 minI와 minJ가 지나온 루트이다.
}
```

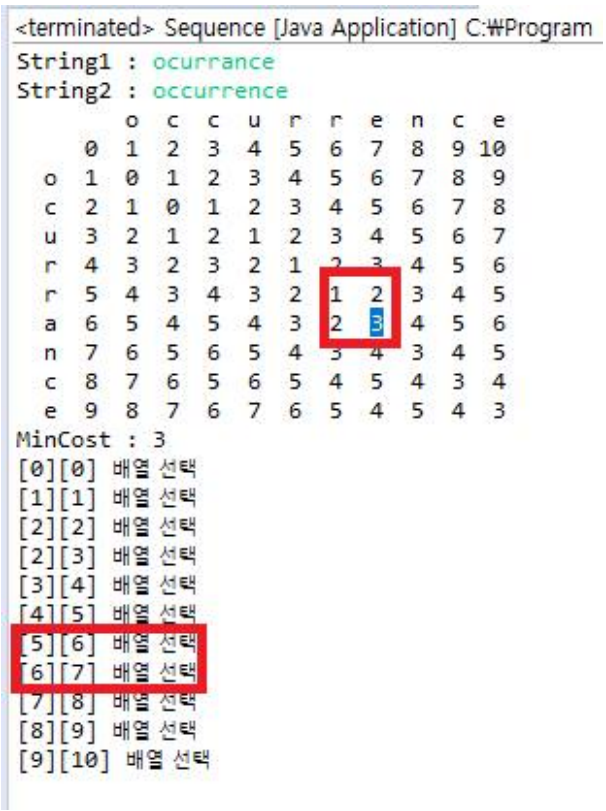
최소값으로 가는 루트를 만들기위해 역순으로 찾아가는 식으로 구현하였다.
역순으로 최소값을 찾아가는식으로 구현하였다.
이렇게 구현하면 gap을 통하여 Match를 구할수있을 경우에는 gap을 하여 가고 더 이상 match가 없을 경우에는 2번의 gap이 아닌 1번의 missmach의 길을 가는 최단 루트를 구할 수 있다.

2. 실행 결과

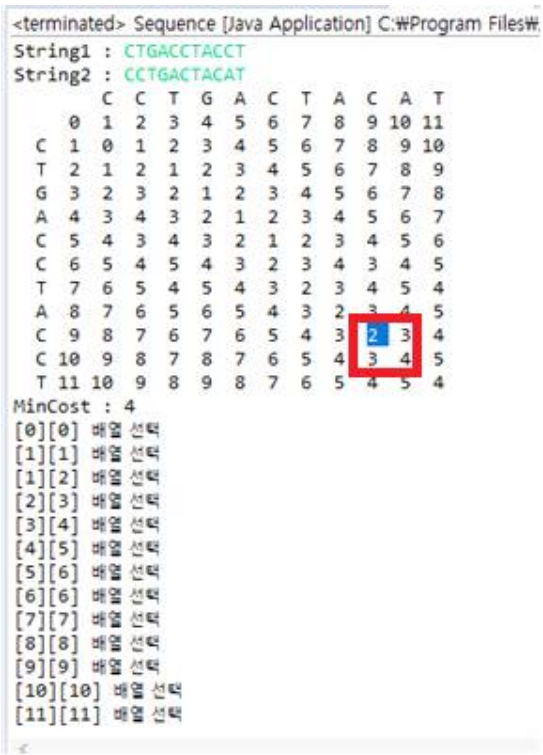


->최적화

를통하여서 gap 2번을 missmatch 1번으로 했다.

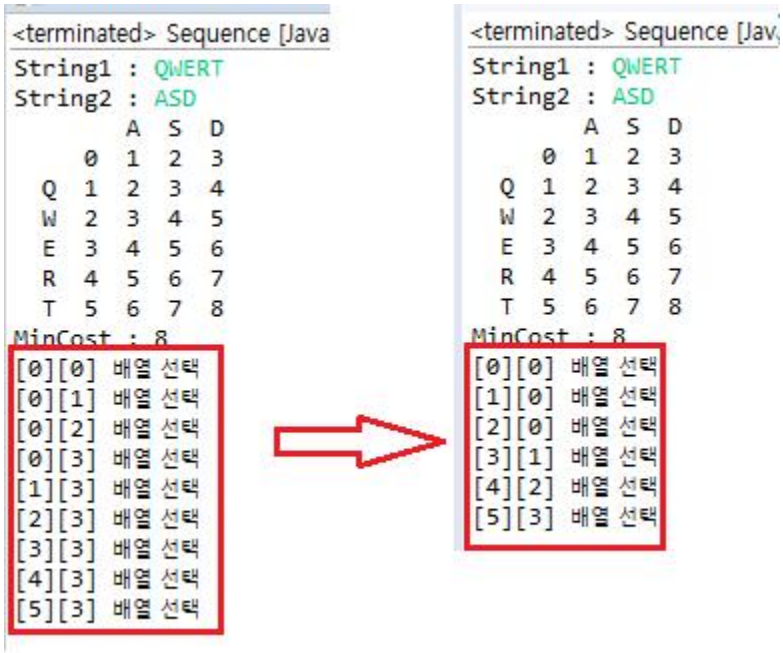


(예제에 나와있는 출력) missmatch 1회로 해도되는데 gap으로 2회로 되어있어서 최소비용은 같지만 길을 경유하여 가고 있다.



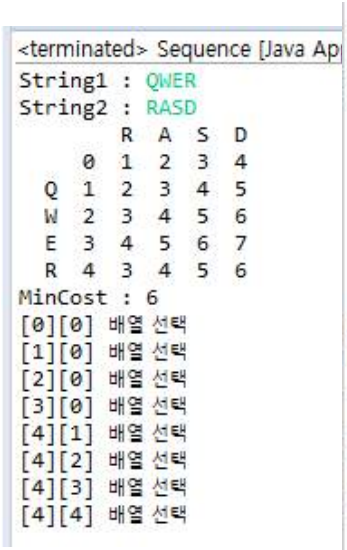
다른예제에 나와있는것도 바꿔준 예시이다

극단적인 차이를 보여주는 예

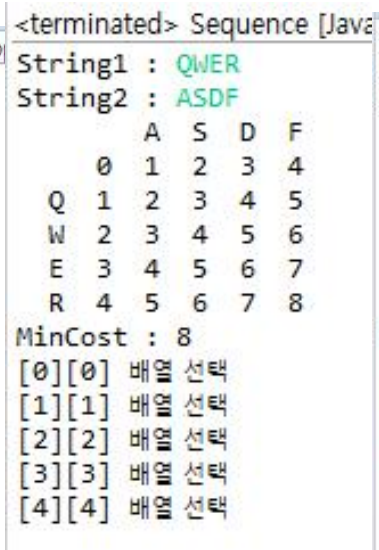


전부다 GAP으로 하면 9회경유로 출력을 하지만
MISSMATCH 3회로 바뀌면 6회만 경유하면된다.

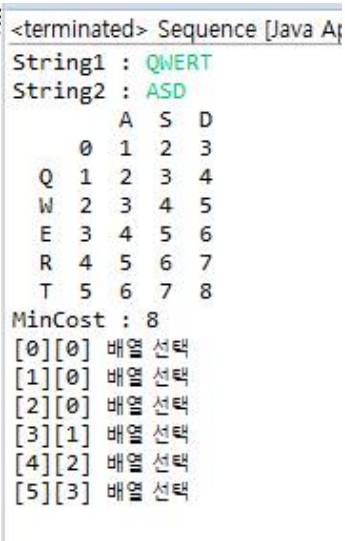
다양한 예로 테스트 해봤더니 잘 실행되었다.



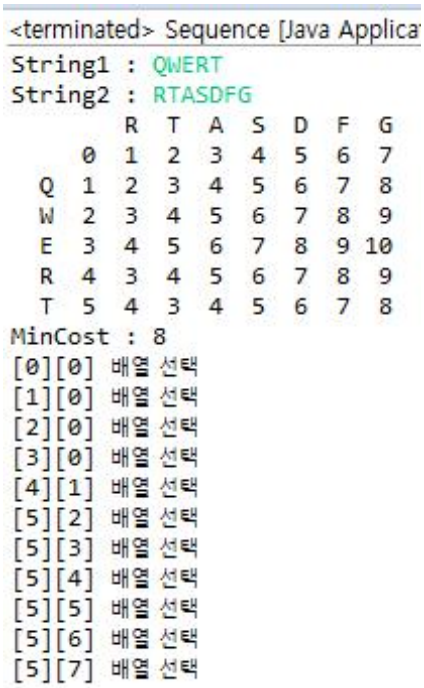
```
<terminated> Sequence [Java Ap
String1 : QWER
String2 : RASD
        R  A  S  D
     0  1  2  3  4
  Q  1  2  3  4  5
  W  2  3  4  5  6
  E  3  4  5  6  7
  R  4  3  4  5  6
MinCost : 6
[0][0] 배열 선택
[1][0] 배열 선택
[2][0] 배열 선택
[3][0] 배열 선택
[4][1] 배열 선택
[4][2] 배열 선택
[4][3] 배열 선택
[4][4] 배열 선택
```



```
<terminated> Sequence [Java
String1 : QWER
String2 : ASDF
        A  S  D  F
     0  1  2  3  4
  Q  1  2  3  4  5
  W  2  3  4  5  6
  E  3  4  5  6  7
  R  4  5  6  7  8
MinCost : 8
[0][0] 배열 선택
[1][1] 배열 선택
[2][2] 배열 선택
[3][3] 배열 선택
[4][4] 배열 선택
```



```
<terminated> Sequence [Java Ap
String1 : QWERT
String2 : ASD
        A  S  D
     0  1  2  3
  Q  1  2  3  4
  W  2  3  4  5
  E  3  4  5  6
  R  4  5  6  7
  T  5  6  7  8
MinCost : 8
[0][0] 배열 선택
[1][0] 배열 선택
[2][0] 배열 선택
[3][1] 배열 선택
[4][2] 배열 선택
[5][3] 배열 선택
```



```
<terminated> Sequence [Java Applica
String1 : QWERT
String2 : RTASDFG
        R  T  A  S  D  F  G
     0  1  2  3  4  5  6  7
  Q  1  2  3  4  5  6  7  8
  W  2  3  4  5  6  7  8  9
  E  3  4  5  6  7  8  9 10
  R  4  3  4  5  6  7  8  9
  T  5  4  3  4  5  6  7  8
MinCost : 8
[0][0] 배열 선택
[1][0] 배열 선택
[2][0] 배열 선택
[3][0] 배열 선택
[4][1] 배열 선택
[5][2] 배열 선택
[5][3] 배열 선택
[5][4] 배열 선택
[5][5] 배열 선택
[5][6] 배열 선택
[5][7] 배열 선택
```

3. 느낀 점

이 과제는 Missmatch와 Gap 2회가 비용이 같아서 애매한 과제인거같다 같지않을 경우에는 항상 gap만 해도 최소비용이되어서 gap1회 < missmath 1회 <gap2회 정도의 가중치를 줬으면 좀 더 생각하는 과제가 되었을거 같다.