LCS : 문자열 두개가 있을 때 최장 공통 부분수열

ACAYKP AGCAT

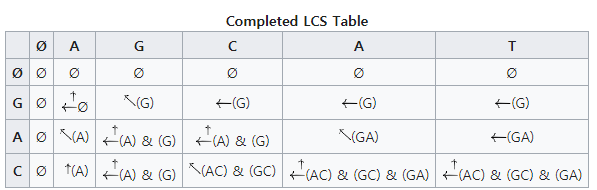
CAPCAK GAC

일 때, 정답은 4 ACAK 다 일때는 2, GA, AC, GC 이다

알고리즘

1. 두 문자열의 길이를 n, m이라고 일 때, (n+1)x(m+1) 길이의 표를 만들고, 첫행과 첫열은 0으로 초기화한다. 그리고 두 문자열을 각 길이에 맞춰 배치한다.(이때 긴 것이 어디에 오든 상관없다)
2. 행으로 배치된 문자열의 첫번째 원소를 열에 배치된 문자열의 원소와 하나씩 비교한다.
3. 같은 것이 나오면 현재칸이 (i, j) 일때 map[i][j] = map[i-1][j-1] + 1
4. 다른 것이 나오면 map[i][j] = max(map[i-1][j], map[i][j-1]) 이다.
5. 이렇게 다하고, 맨 마지막 행, 마지막 열부터 차례로 검사하면 최장공통부분수열을 구할 수 있다.
6. 마지막 행의 마지막 열부터 검사하다가, 옆 열과 1이 차이나고 열과 행에 있는 문자 두개가 같으면 그 문자를 LCS 배열에 넣고 왼쪽 위로 이동하여, 똑같이 실시. 이때 이전에 넣었을 때 맵에 있던 값과 다음에 넣을 때 맵에 있는 값은 1만 차이나야한다.
7. 이렇게 해서 첫번째 열까지 이동하면 종료 후, LCS 길이 저장하면 됨.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | **A** | **C** | **A** | **Y** | **K** | **P** |
|  | **0** | **0** | **0** | **0** | **0** | **0** | **0** |
| **C** | **0** | **0** | **1** | **1** | **1** | **1** | **1** |
| **A** | **0** | **1** | **1** | **2** | **2** | **2** | **2** |
| **P** | **0** | **1** | **1** | **2** | **2** | **2** | **3** |
| **C** | **0** | **1** | **2** | **2** | **2** | **2** | **3** |
| **A** | **0** | **2** | **2** | **3** | **3** | **3** | **3** |
| **K** | **0** | **2** | **2** | **3** | **3** | **4** | **4** |



모든 LCS가 필요할 때는 이렇게하자