

28_다이나믹프로그래밍_크리보드

_LCS_LSC2

1. 크리보드

문제

4개의 버튼이 있는 키보드를 눌러 화면에 출력된 A개수의 최댓값 구하기

로직

1. 화면에 A를 출력하는 경우: $dp[i] = dp[i - 1] + 1$
 2. Ctrl-A → Ctrl-C → Ctrl-V를 누르는 경우
 - 마지막에 Ctrl-V를 한 번 누르는 경우: $dp[i] = dp[i - 3] * 2$
 - 마지막에 Ctrl-V를 두 번 누르는 경우: $dp[i] = dp[i - 4] * 3$
 -
 - 마지막에 Ctrl-V를 j번 누르는 경우: $dp[i] = dp[i - (j+2)] * (j + 1)$
- 이 때, $1 \leq j \leq i - 2$

1, 2번의 경우를 비교하여 최댓값을 구할 수 있다.

시간 복잡도

$O(N^2)$

2. LCS

문제

두 문자열 A와 B가 주어졌을 때 가장 긴 공통 부분수열의 길이를 구하는 문제

로직

두 문자열에 적절히 공백을 삽입하여 같은 위치에 같은 문자가 있도록 한다.

$dp[i][j]$: A의 길이가 i, B의 길이가 j일 때 LCS의 길이

1. A의 i번째 문자 == B의 j번째 문자: $dp[i][j] = dp[i - 1][j - 1] + 1$
2. A의 i번째 문자 != B의 j번째 문자
 - 1) A의 마지막에 공백을 삽입하는 경우: $dp[i][j] = dp[i][j - 1]$
 - 2) B이 마지막에 공백을 삽입하는 경우: $dp[i][j] = dp[j - 1][i]$

코드

```
a = " " + a;
b = " " + b;
```

위 부분을 통해 for문의 인덱스를 1부터 사용할 수 있다.

for문 내부의 코드 $dp[i-1][j-1]$ 에서 i,j가 0인 경우 인덱스가 배열의 범위를 넘어가는 문제가 발생하는데, 위 코드를 통해이 문제가 생기는 것을 방지할 수 있다.

3. LCS2

문제

두 문자열 A와 B가 주어졌을 때 가장 긴 공통 부분수열의 길이와 공통 문자를 구하는 문제

로직

문자열을 출력하기 위해 부분 수열의 길이를 구한 뒤, 문자열을 역추적해나간다.