LIS, LCS : 가장 긴 증가하는 부분 수열

(Longest Increasing Subsequence)

(Longest Common Subsequence)

1. 가장 긴 증가하는 부분 수열 ( #11053, #2565 )

수열이 주어집니다.

수열의 가장 긴 증가하는 부분 수열의 길이를 계산하시오.

**입력**

**6**

**10 20 10 30 20 50**

**출력**

**4**

**"10 20 30 50”**

1. 두 수열에 포함되는 가장 긴 부분 문자열 ( #9251 )

두 문자열이 주어집니다.

양 쪽 문자열에 공통적으로 포함되는 가장 긴 부분 문자열의 길이를 계산하시오.

**입력**

**ACAYKP**

**CAPCAK**

**출력**

**4**

**“ACAK”**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | C | A | P | C | A | K |
| A |  |  |  |  |  |  |
| C |  |  |  |  |  |  |
| A |  |  |  |  |  |  |
| Y |  |  |  |  |  |  |
| K |  |  |  |  |  |  |
| P |  |  |  |  |  |  |

**정답코드**

**#11053**

n = int(input())

arr = list(map(int,input().split()))

dp = [1 **for** \_ **in** range(n)]

**for** i **in** range(n):

**for** j **in** range(i):

**if** arr[i] > arr[j]:

dp[i] = max(dp[i], dp[j] + 1)

print(max(dp))

**#9251**

a = list(str(input()))

b = list(str(input()))

dp = [[0 **for** \_ **in** range(len(b)+1)]**for** \_ **in** range(len(a)+1)]

**for** i **in** range(1,len(a) +1):

**for** j **in** range(1,len(b) +1):

**if** a[i-1] == b[j-1]:

dp[i][j] = dp[i-1][j-1]+ 1

**else**:

dp[i][j] = max(dp[i-1][j], dp[i][j-1])

*# print(dp)*

print(dp[len(a)][len(b)])