

{% note info %} **摘要** Title: 897. 最长公共子序列 Tag: dp、LCS、dp注意事项 Memory Limit: 64 MB Time Limit: 1000 ms {% endnote %}

Powered by:NEFU AB-IN

[Link](#)

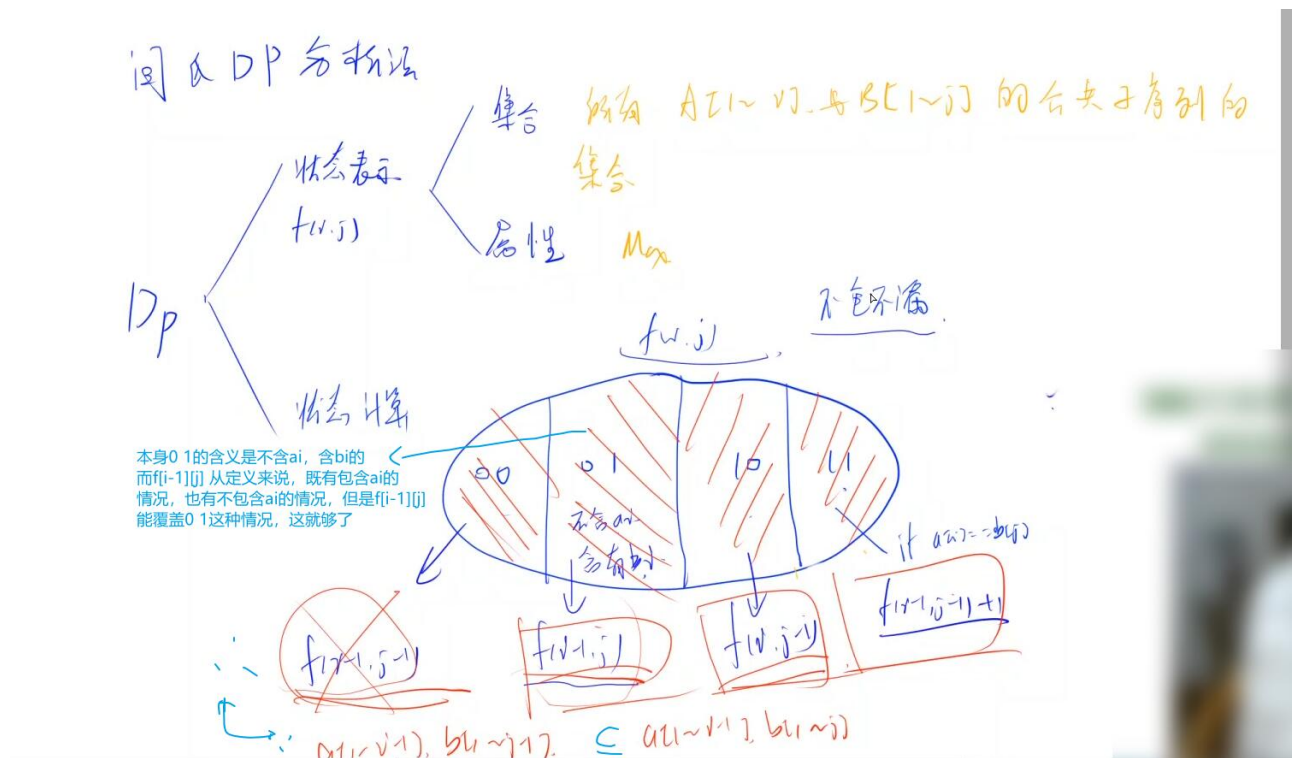
@TOC

897. 最长公共子序列

• 题意

给定两个长度分别为 N 和 M 的字符串 A 和 B ，求既是 A 的子序列又是 B 的子序列的字符串长度最长是多少。

• 思路



ps:

- 这只是求最大值可以覆盖，求和就不行了
- 自己写**状态表示**时， i 不单单代表 i 这个点，更多可能是 $a[1 \sim i]$ 这个区间
- 一般题目问的什么，集合就是什么
- **状态计算**，一般看**最后一步**怎么走，比如
 - **背包问题**：最后一个物品选择了几个（01背包就是是否选）
 - **LCS**：最后两个元素相等不相等
 - **LIS**：倒数第二个元素选的是哪个

• 代码

```
'''
Author: NEFU AB-IN
Date: 2022-03-07 16:21:35
FilePath: \ACM\Acwing\897.py
LastEditTime: 2022-03-07 16:21:36
'''

N, M = 1100, 1100

dp = [[0] * M for _ in range(N)]

n, m = map(int, input().split())

a = " " + input()
b = " " + input()

for i in range(1, n + 1):
    for j in range(1, m + 1):
        dp[i][j] = max(dp[i - 1][j], dp[i][j - 1])
        if a[i] == b[j]:
            dp[i][j] = max(dp[i][j], dp[i - 1][j - 1] + 1)

print(dp[n][m])
```