LeetCode

< ≡ Problem List >

Premium

Description | 🔒 Editorial | Solutions (3.6K) | Submissions

× Close

✅ Accepted

723319809
Mar 30, 2023 21:56

Details | + Solution

Python3

Next question

• 1147. Longest Chunked Palindrome Decomposition

Runtime **1122 ms** Beats **22.13%** Memory **48.7 MB** Beats **18.88%**

Click the distribution chart to view more details

More challenges

• 516. Longest Palindromic Subsequence   • 583. Delete Operation for Two Strings   • 1092. Shortest Common Supersequence
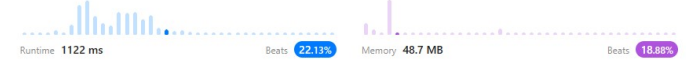
Notes

Write your notes here

Related Tags

Select tags   0/5

All statuses ▾   All languages ▾

Accepted
a few seconds ago   Python3   >

```python
class Solution:
    def longestCommonSubsequence(self, text1: str, text2: str) -> int:
        m=len(text1)+1
        n=len(text2)+1
        b=[[-1 for i in range(0,n)] for j in range(0,m)]
        c=[[-1 for i in range(0,n)] for j in range(0,m)]
        LCSLength(m,n,text1,text2,c,b)
        return c[m-1][n-1]

def LCSLength(m:int, n:int, x:list, y:list, c:list[list],b:list[list]):
    for i in range(0,m):
        c[i][0]=0
    for j in range(0,n):
        c[0][j]=0
    for i in range(1,m):
        for j in range(1,n):
```

```python
class Solution:
    def longestCommonSubsequence(self, text1: str, text2: str) -> int:
        m=len(text1)+1
        n=len(text2)+1
        b=[[-1 for i in range(0,n)] for j in range(0,m)]
        c=[[-1 for i in range(0,n)] for j in range(0,m)]
        LCSLength(m,n,text1,text2,c,b)
        return c[m-1][n-1]


def LCSLength(m:int, n:int, x:list, y:list, c:list[list],b:list[list]):
    for i in range(0,m):
        c[i][0]=0
    for j in range(0,n):
        c[0][j]=0
    for i in range(1,m):
        for j in range(1,n):
            if x[i-1]==y[j-1]:
                c[i][j]=c[i-1][j-1]+1
                b[i][j]=1
            elif c[i-1][j]>=c[i][j-1]:
                c[i][j]=c[i-1][j]
                b[i][j]=2
            else:
                c[i][j]=c[i][j-1]
                b[i][j]=3
def LCS(i:int, j:int, x:list, b:list[list]):
    if i==0 or j==0:
        return
    if b[i][j]==1:
        LCS(i-1,j-1,x,b)
        print(x[i])
    elif b[i][j]==2:
        LCS(i-1,j,x,b)
    else:
        LCS(i,j-1,x,b)


if __name__ == '__main__':
    text1='ABCBDAB'
    text2='BDCABA'
    test=Solution()
    print(test.longestCommonSubsequence(text1,text2))
```