

# Longest Common Substring (En Uzun Ortak Alt Dizi)

## \*\*Tanım:\*\*

İki string arasında en uzun ortak kesintisiz alt diziyi bulma problemdir. Yani karakterler ardı ik olarak aynı olmalıdır. Bu nedenle, 'substring' kavramı 'subsequence'den farklıdır.

## \*\*DP Mantıtı\*\*

Bir DP (Dynamic Programming) tablosu oluşturulur.  $dp[i][j]$  de eri, A[i-1] ve B[j-1] karakterleri eşitse,  $dp[i][j] = dp[i-1][j-1] + 1$  olur. Eşit değilse,  $dp[i][j] = 0$  yapılır. Sonuç, tablodaki maksimum değerdir.

## \*\*DP Formülü:\*\*

$$A[i-1] == B[j-1] \rightarrow dp[i][j] = dp[i-1][j-1] + 1$$

$$A[i-1] \neq B[j-1] \rightarrow dp[i][j] = 0$$

## \*\*Örnek:\*\*

A = 'abcdef'

B = 'abedf'

		a	b	e	d	f
	0	0	0	0	0	0
a	0	1	0	0	0	0
b	0	0	2	0	0	0
c	0	0	0	0	0	0
d	0	0	0	0	1	0
f	0	0	0	0	0	2

## \*\*DP Tablosu Görselleştirme:\*\*

A[i]	B[j]	dp[i-1][j-1]	dp[i][j]	max_len
------	------	--------------	----------	---------

### \*\*Python Kodu:\*\*

```
def longest_common_substring(a, b):
    n, m = len(a), len(b)
    dp = [[0]*(m+1) for _ in range(n+1)]
    max_len = 0
    end_index = 0
    for i in range(1, n+1):
        for j in range(1, m+1):
            if a[i-1] == b[j-1]:
                dp[i][j] = dp[i-1][j-1] + 1
                if dp[i][j] > max_len:
                    max_len = dp[i][j]
                    end_index = i
            else:
                dp[i][j] = 0
    return a[end_index - max_len:end_index]
```

### \*\*Sonuç:\*\*

- Substring kesintisiz olmalı.
- DP yaklaşımlı hızlı ve verimlidir.
- Tablo maksimum değer üzerinden sonucu verir.
- Gerçek dünyada metin ve DNA analizi gibi alanlarda kullanılır.