Dynamic Programming

Video - 59

Note :- This playlist is only for explanation of Qns & solutions.

See my DP Concepts & Qns playlist for understanding DP from Scratch...

codestorywithMIK

Facebook, Instagram → codestorywithMIK

Twitter → cswithMIK

Leetcode - 97

(Medium) → Easy | Simple Recursion + Memoization

- 3 variables memoization
- 2 variables memoization

Interleaving String :-

Company :- Google FACTSET yahoo! Zillious

97. Interleaving String

Medium 6759 408 Add to List Share

Given strings `s1`, `s2`, and `s3`, find whether `s3` is formed by an **interleaving** of `s1` and `s2`.

An **interleaving** of two strings `s` and `t` is a configuration where `s` and `t` are divided into `n` and `m` substrings respectively, such that:

- $s = s_1 + s_2 + ... + s_n$
- $t = t_1 + t_2 + ... + t_m$
- $|n - m| <= 1$
- The **interleaving** is $s_1 + t_1 + s_2 + t_2 + s_3 + t_3 + ...$ or $t_1 + s_1 + t_2 + s_2 + t_3 + s_3 + ...$

Note: `a + b` is the concatenation of strings `a` and `b`.

s3 = aadbbcbcac

s3 = aa dbbc bc a c

s1 = aa bc c

s2 = d bb c a

aa
bc
c

aa dbbc bc a c
s1 s2 s1 s2 s1

dbbc
a

Why DP?

(s1.length() + s2.length() != s3.length()) return False;

s3 = a a d b b c b c a c
K
0 1 2 3 4 5 6 7 8 9
N

s1 = m
s2 = n
s3 = N

s1 = a a b c c
i
0 1 2 3 4
m

s2 = d b b c a
j
0 1 2 3 4
n

s3 = N

s1 = a a b c c c (i, m; 0 1 2 3 4)

s2 = a d b b c a (j, n; 0 1 2 3 4)

aadbbcbcac

→ { if (i == m && j == n && K == N)
    return True;

→ { if (K >= N)
    return False;

s3 = " a a b " (K)

s1 = " a a b " (i)

s2 = " " (j)

~~i == m && j == N && K == N~~

if (K == N)

if (K == N)

see (alse)

S3 = "aab" K

S1 = "aab" i / 3

S2 = "" j / 0

options:-

S1 ith

S2 jth

Solve(i, j) {

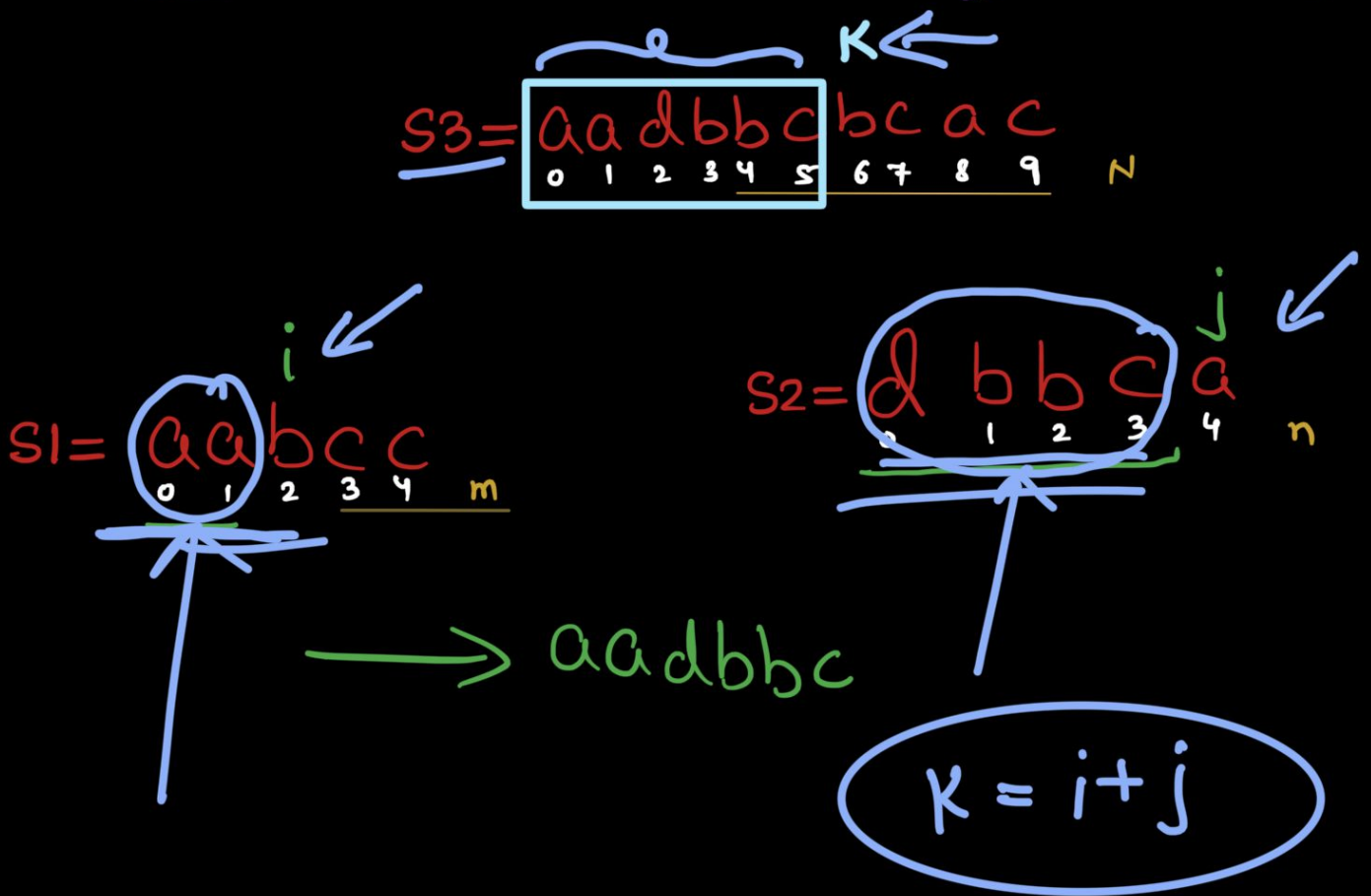
if (i == m && j == n && (i+j) K == N)

```
Solve( i, j ) {
    if (i==m && j==n && (i+j)==N)
        return True;
    if (i+j >= N)
        return False;

    if (S1[i] == S3[i+j])
        result = solve( i+1, j );
    if (S2[j] == S3[i+j])
        result = solve( i, j+1 );

    return result;
}
```

Memoization:

t[101][101][101];

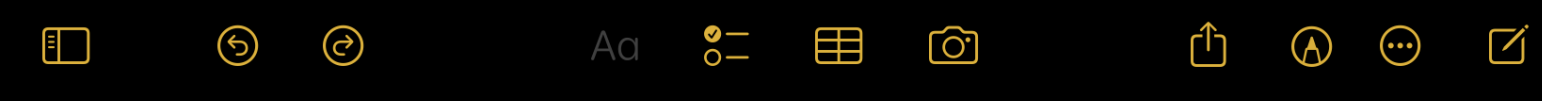
2-Variables Memo*



[i] [j]

[101][101]

O(m*n)



[i] [j]

[101][01]

O(m*n)