Dynamic Programming

Video - 92

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

See my "DP Concepts & Qns" playlist for understanding DP from scratch...

Facebook, Instagram → codestorywithMIK

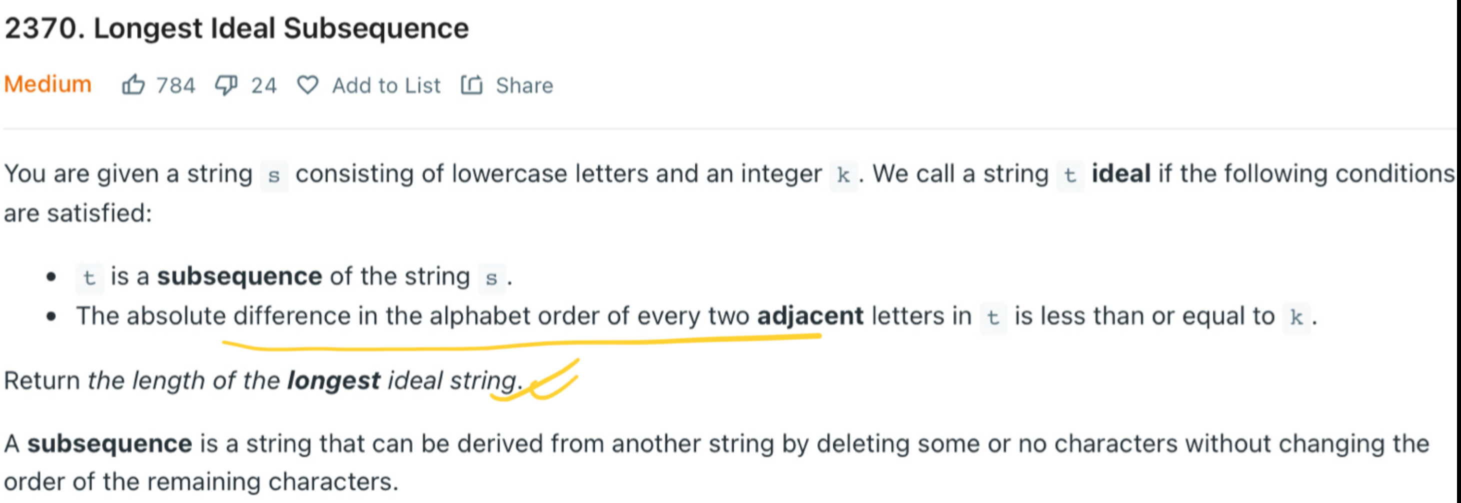
Twitter → cswithMIK

☎ → codestorywithMIK

Leetcode - 2370

Medium

→ Simple LIS variant :)



2370. Longest Ideal Subsequence

Medium 👍 784 👎 24 ♡ Add to List Share

You are given a string `s` consisting of lowercase letters and an integer `k`. We call a string `t` **ideal** if the following conditions are satisfied:

- `t` is a **subsequence** of the string `s`.
- The absolute difference in the alphabet order of every two **adjacent** letters in `t` is less than or equal to `k`.

Return *the length of the* ***longest*** *ideal string*. ✓

A **subsequence** is a string that can be derived from another string by deleting some or no characters without changing the order of the remaining characters.

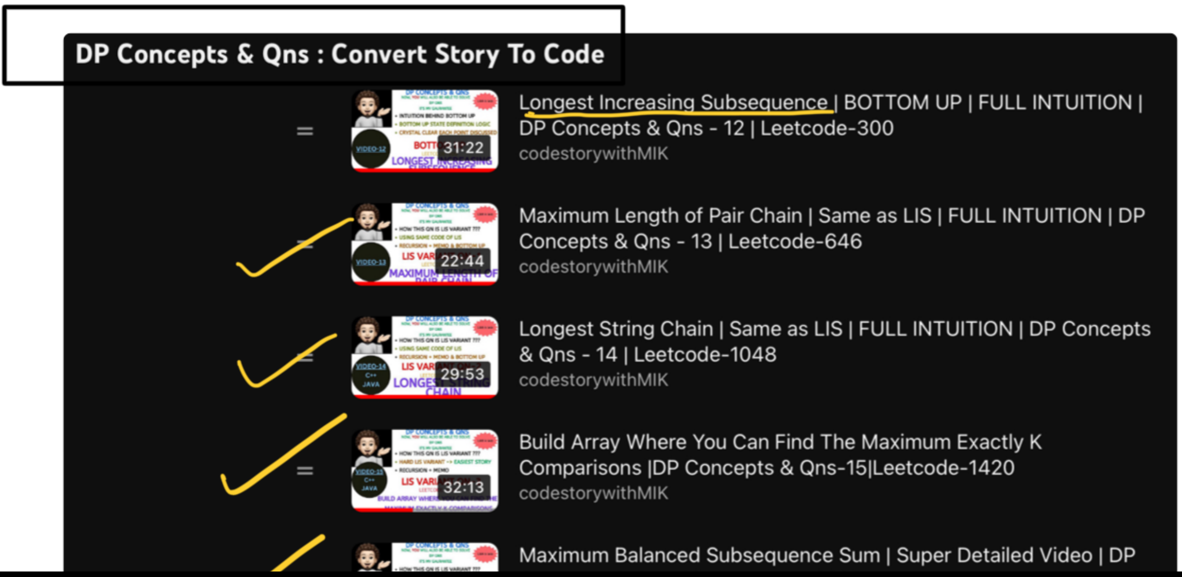
Example:- s = "acfgbd" , K = 2

Output = 4   t = "acbd"

s = "abcd" , K = 3

Output = 4   t = "abcd"

<= K



DP Concepts & Qns : Convert Story To Code

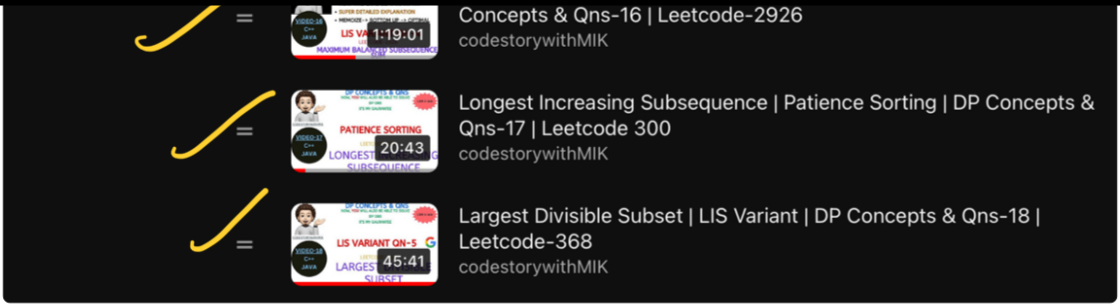
31:22 — Longest Increasing Subsequence | BOTTOM UP | FULL INTUITION | DP Concepts & Qns - 12 | Leetcode-300
codestorywithMIK

✓ 22:44 — Maximum Length of Pair Chain | Same as LIS | FULL INTUITION | DP Concepts & Qns - 13 | Leetcode-646
codestorywithMIK

✓ 29:53 — Longest String Chain | Same as LIS | FULL INTUITION | DP Concepts & Qns - 14 | Leetcode-1048
codestorywithMIK

✓ 32:13 — Build Array Where You Can Find The Maximum Exactly K Comparisons |DP Concepts & Qns-15|Leetcode-1420
codestorywithMIK

✓ Maximum Balanced Subsequence Sum | Super Detailed Video | DP



Concepts & Qns-16 | Leetcode-2926
codestorywithMIK

Longest Increasing Subsequence | Patience Sorting | DP Concepts & Qns-17 | Leetcode 300
codestorywithMIK

Largest Divisible Subset | LIS Variant | DP Concepts & Qns-18 | Leetcode-368
codestorywithMIK

Intuition :-

LIS :-

0	1	2	3	4
{5},	{4},	{2},	{3},	{6}
↑	↑	↑		
{5}	{4}	{2}	{2,3}	{2,3,6}

→ 2,3

{2,3,6}

3

nums[j] < nums[i]

cr {a}	{a,c}		{f,g}	{a,c,b}	{a,c,b,d}
a	c	f	g	b	d
0	1	2	3	4	5

, K = 2

i

t	1	2	1	2	3	4

|a-c| <= K

for (i = 0; i < n; i++){

```
for (j = i-1; j>=0; j--) {
        if (|char-i - char-j| <= k) {
              t[i] = max(t[i], t[j]+1)
}       }
```

LIS

$T.C = O(n^2)$

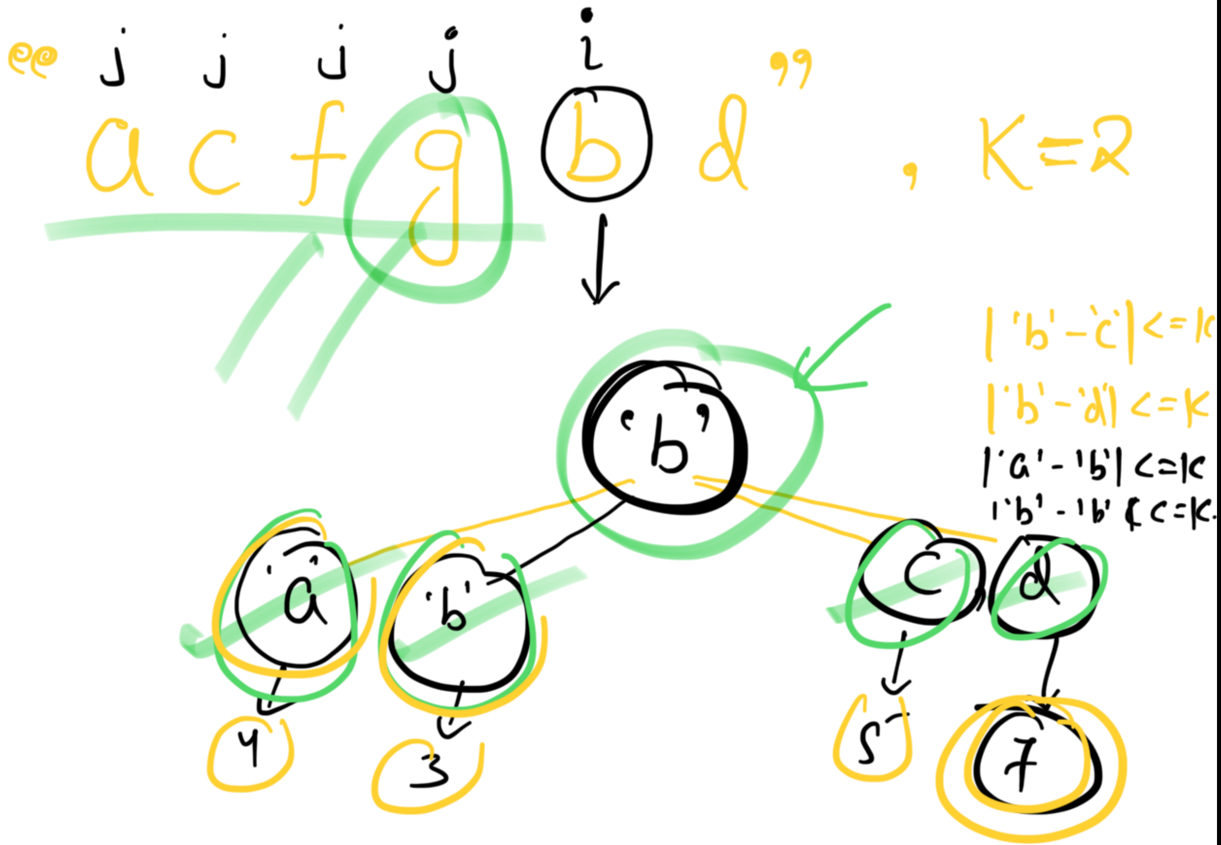
$S.C = O(n)$

// t[i] = LIS ending at index i

ideal

Optimise...

"acfgbd", K=2



|'b'-'c'|<=k

|'b'-'d'|<=K

|'a'-'b'|<=k

|'b'-'b'{<=k.

"a c f g b d", K=2

i

(ac)

(acb) (acb d)

(fg)

```
int idx = S[i] - 'a'; // 3
int left = max (0, idx-K); // 1
int right = min (25, idx+K); // 5
```

0	1	2	3	4	5	6	7	8	25
1	3	2	4	0	1	2	0	0 . . .	0
a	b	c	d	e	f	g	h	. . .	z

longest = 3 +1 = 4