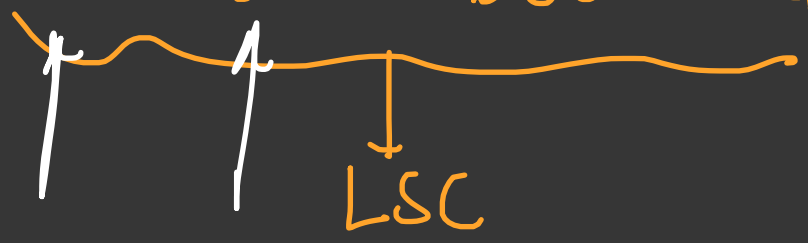
Longest String Chain

s =

```
words[]: [ "a", "b", "ba", "bca", "bda", "bdca" ]
```

LSC? → a b a bda bdca



Why? → "a" + "b" → then it is considered in a chain.

append

~~Now~~ prev word + new char ⟶ Include in chain

Let compare it with LIS ?

⟶ In LIS ⟶ we check ~~if~~ ~~to~~ my prev element is smaller than current.

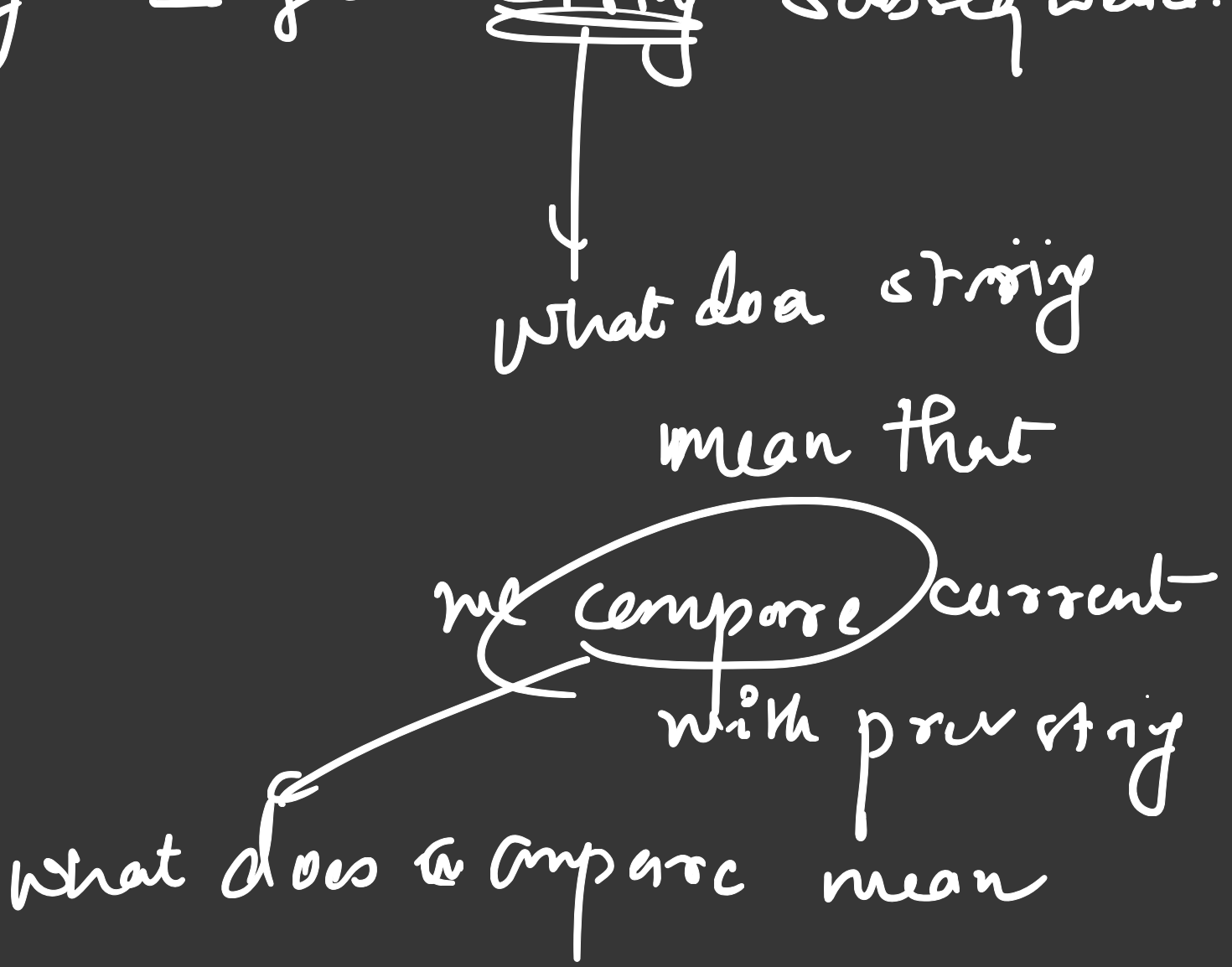
1 4 3 7 8

1 < 4 < 7 < 8

1 4 7 8 ⟶ LIS

↑

So, I can say Longest String Subsequence.



Comparing a String means → all the char. of

prev should be there

and a additional one

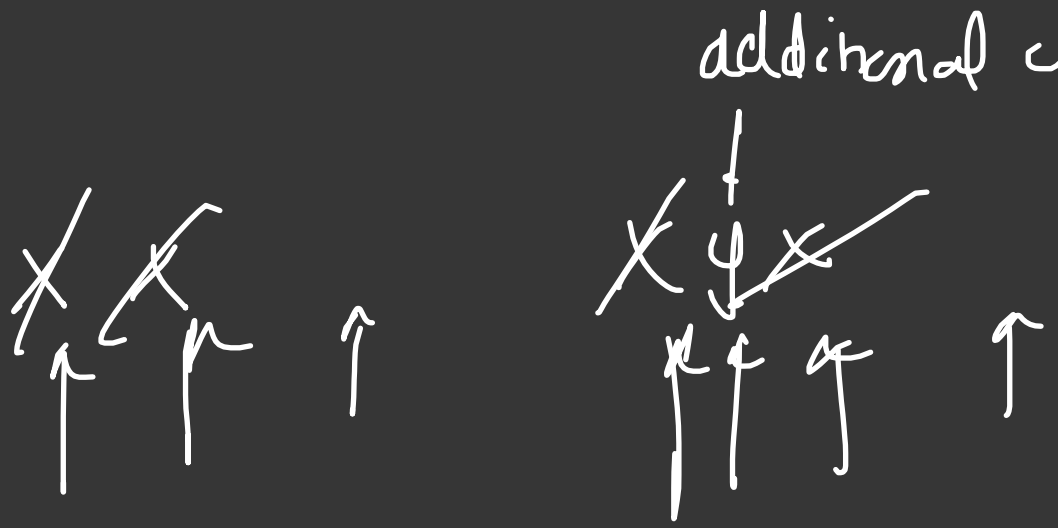
char.

and it size should be 1 greater.

Pseudocode:

```
for ( i = 1, i < n; i++ )
    for ( j = 0, j < i; j++ )
        if ( Compare(s[i], s[j]) )
            dp[i] = max ( dp[i], 1 + dp[j] )
    maxi = max ( maxi, dp[i] );
```

Compare function:



True