@codestorywithmik
(Instagram, Facebook)
cswithMIK → Twitter
codestorywithMIK → whatsapp

STRINGS...

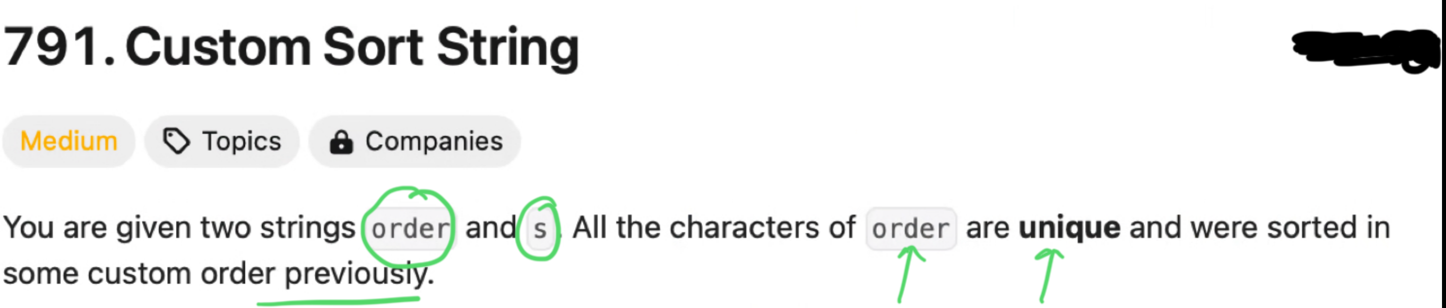
Video-35

Leetcode-791

~~Medium~~

2 Approaches

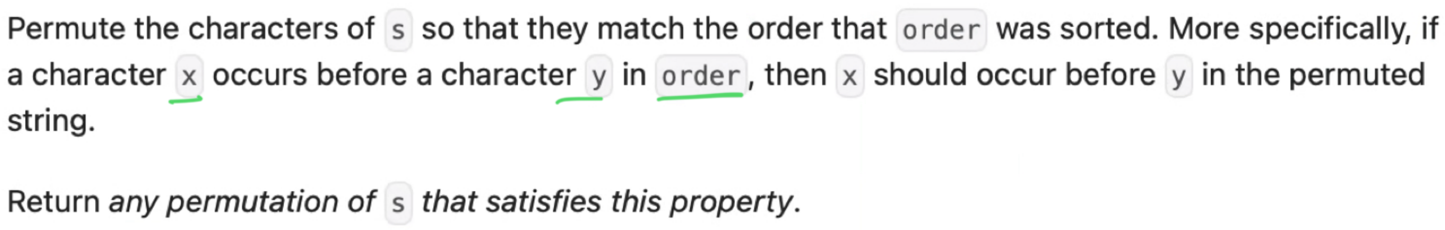
Asked By :- Meta



791. Custom Sort String [redacted]

Medium · Topics · Companies

You are given two strings `order` and `s`. All the characters of `order` are **unique** and were sorted in some custom order previously.



Permute the characters of `s` so that they match the order that `order` was sorted. More specifically, if a character `x` occurs before a character `y` in `order`, then `x` should occur before `y` in the permuted string.

Return *any permutation of* `s` *that satisfies this property*.

Example :- order = "cba"

s = "abcd"

Output = "cbad"

Order = "bcafg"

s = "abcd"

Output = "bcad"

÷ Focus on my Thought Process...

Approach-1

(Simplest)

m ← Order = "cba"

n ← s = "abcdadde"

result = "Cbaaddde"

map O(1)

a	0
b	0
c	0
d	0
e	0

result = cbaaddde

T.C = O(m+n)

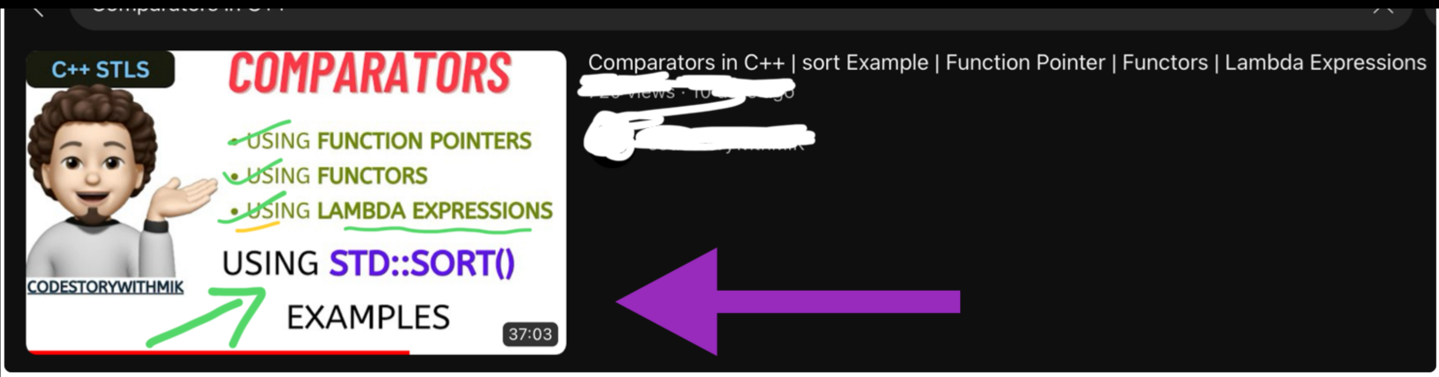
S.C = O(n) → map

26 letter

arr[26]

Approach-2

Comparators in C++



Comparators in C++ | sort Example | Function Pointer | Functors | Lambda Expressions

[redacted]

[redacted]

m

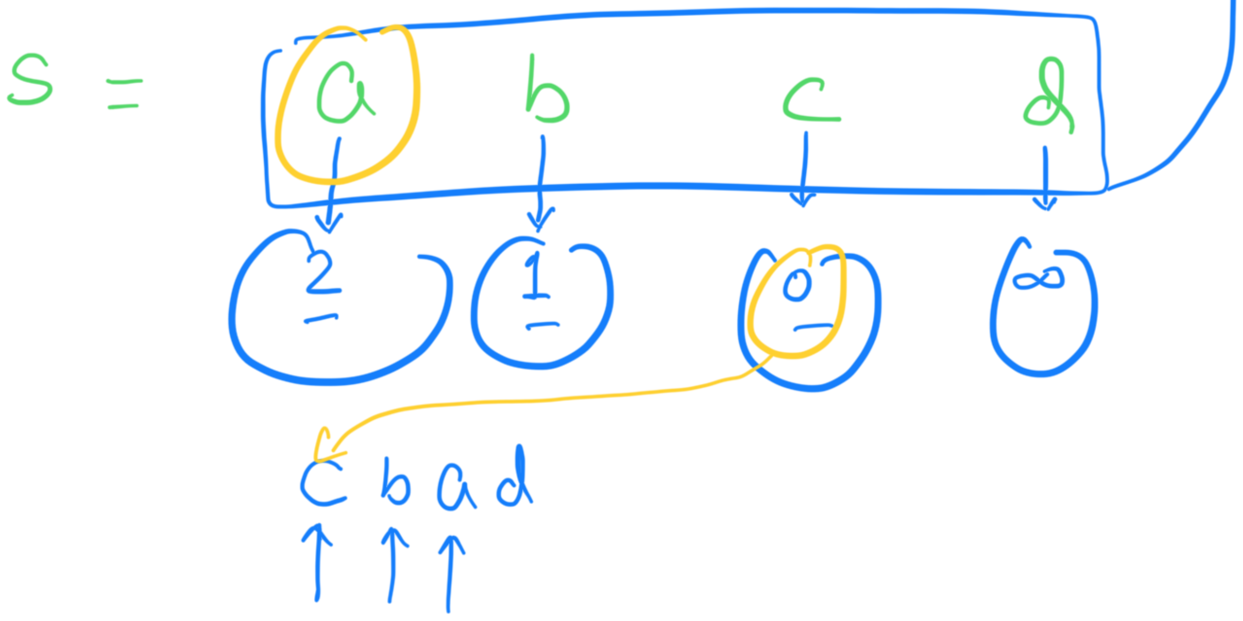
Order = "$\overset{0}{c}\,\overset{1}{b}\,\overset{2}{a}$"

s = "a b c d"

map(

c	0
b	1
a	2

arr(26)



Custom_Sort:-

Sort (begin (s), end (s)); // inc..

```
Sort (begin(s), end(s), myComparator);
```

T.C = O(nlogn)

S.C = O(26)

≃ O(1).

```
auto myComparator = [&map] (char &ch1, char&ch2) {

        if (mp[ch1] < mp[ch2])
                return True;

        else
                retn False;

};
```

```
return
  mp[ch1] < mp[ch2];
```