

Problem: Given 2 strs, S and T return if they equal when both are typed out. Any '#' counts as backspace (# deletes the previous character)

"cb#d" -> cd

S: "ab#c" -> ac

T: "az#c" -> ac

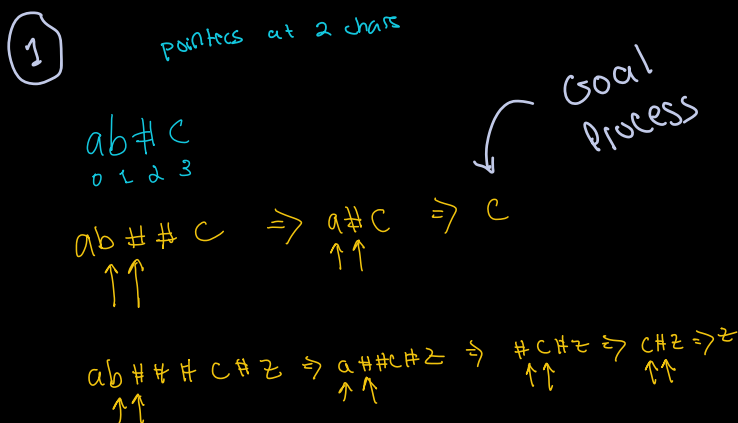
return True

Intro:

- Verify Constraints
 - 2 #'s in a row? "ab##" -> ""
 - #'s if no character to remove: "a####b" -> "b"
 - Are two empty strings equal to each other -> yes "" == ""
 - case sensitivity matters? yes "a" != "A"
- Create Testcases

Brute Force:

- Brainstorming & Pattern Observations



My
Brute force:
2 pointers

Example:

ac#ere####z#a
0 1 2 3 4 5 6 7 8 9 10 11

1st
ac#
0 1 2
↑↑

2nd
ac#
0 1 2
↑↑

$s[0:p+1] + s[p+1:]$

⇒ don't move
the pointers

3rd
aere#
↑↑

4th
aere
↑↑

5th
aere####
0 1 2 3 4
↑↑

6th
aer####z
0 1 2 3 4 5 6
↑↑

shift the
pointers
down!

74
a e r # # # z
↑ ↑
0 1 2 3
g h

a e # # z
0 1 2 3 4
↑ ↑
shift pointers

② MTCI

For str1 and str2

add char to array
if char #, then delete last entry in array

compose 2 arrays

- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity

◦ My Brute Force:

Time: $O(a+b) = O(a) + O(b)$
Space: $O(1)$

only pointers used

iterate thru whole array for str 1
iterate thru whole array for str 2

- optimal space already

◦ MTCI:

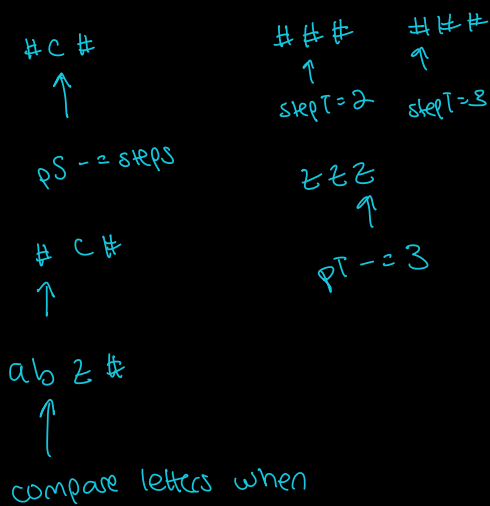
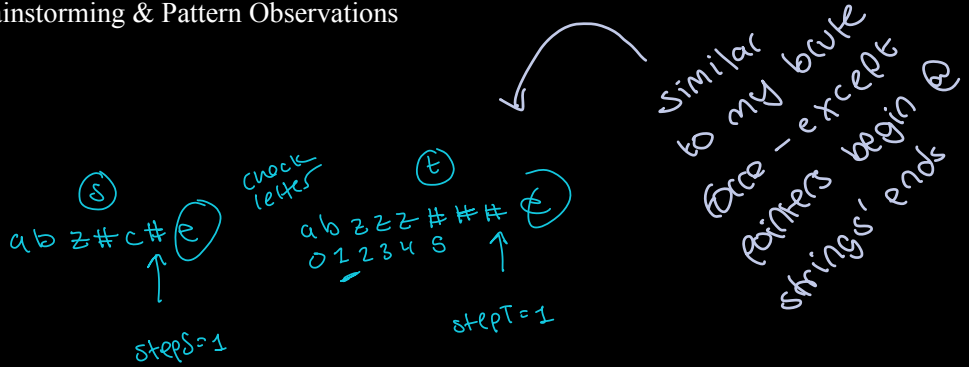
Time: $O(a+b) = O(a) + O(b)$
Space: $O(a+b) + O(a) + O(b)$

b/c of storing the resulting string in an array

- not optimal space

Optimal:

- Brainstorming & Pattern Observations



$stepS == 0$ and $stepT == 0$

and $s[stepS] != '#'$

end

$t[stepT] != '#'$

- Pseudocode
- Write code
- Run through testcases
- Analyze time and space complexity
 - Time: $O(a+b)$
 - Space: $O(1)$

