

Introduction

(medium)

LinkedList Cycle (easy)

Start of LinkedList Cycle

Happy Number (medium)

for Coding Questions

Search Course

(medium)

(medium)

(medium)

Challenge 1

Challenge 2

a Target (medium)

Problem Challenge 1

Problem Challenge 2

Solution Review: Problem

Solution Review: Problem

Squaring a Sorted Array (easy)

Triplet Sum to Zero (medium)

Triplet Sum Close to Target

Triplets with Smaller Sum

Subarrays with Product Less than

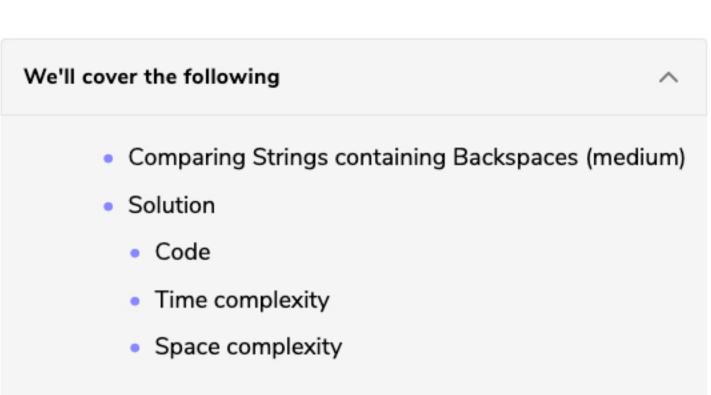
**Dutch National Flag Problem** 

16% completed

Problem Challenge 1

Middle of the LinkedList (easy)

#### Solution Review: Problem Challenge 2



#### Comparing Strings containing Backspaces (medium)

Given two strings containing backspaces (identified by the character '#'), check if the two strings are equal.

₩

? Ask a Question

#### Example 1:

```
Input: str1="xy#z", str2="xzz#"
Output: true
Explanation: After applying backspaces the strings become "xz" and "xz" respectively.
Example 2:
```

#### Input: str1="xy#z", str2="xyz#"

Output: false

Output: true

```
Explanation: After applying backspaces the strings become "xz" and "xy" respectively.

Example 3:
Input: str1="xp#", str2="xyz##"
```

In "xyz##", the first '#' removes the character 'z' and the second '#' removes the characte

Explanation: After applying backspaces the strings become "x" and "x" respectively.

#### r 'y'.

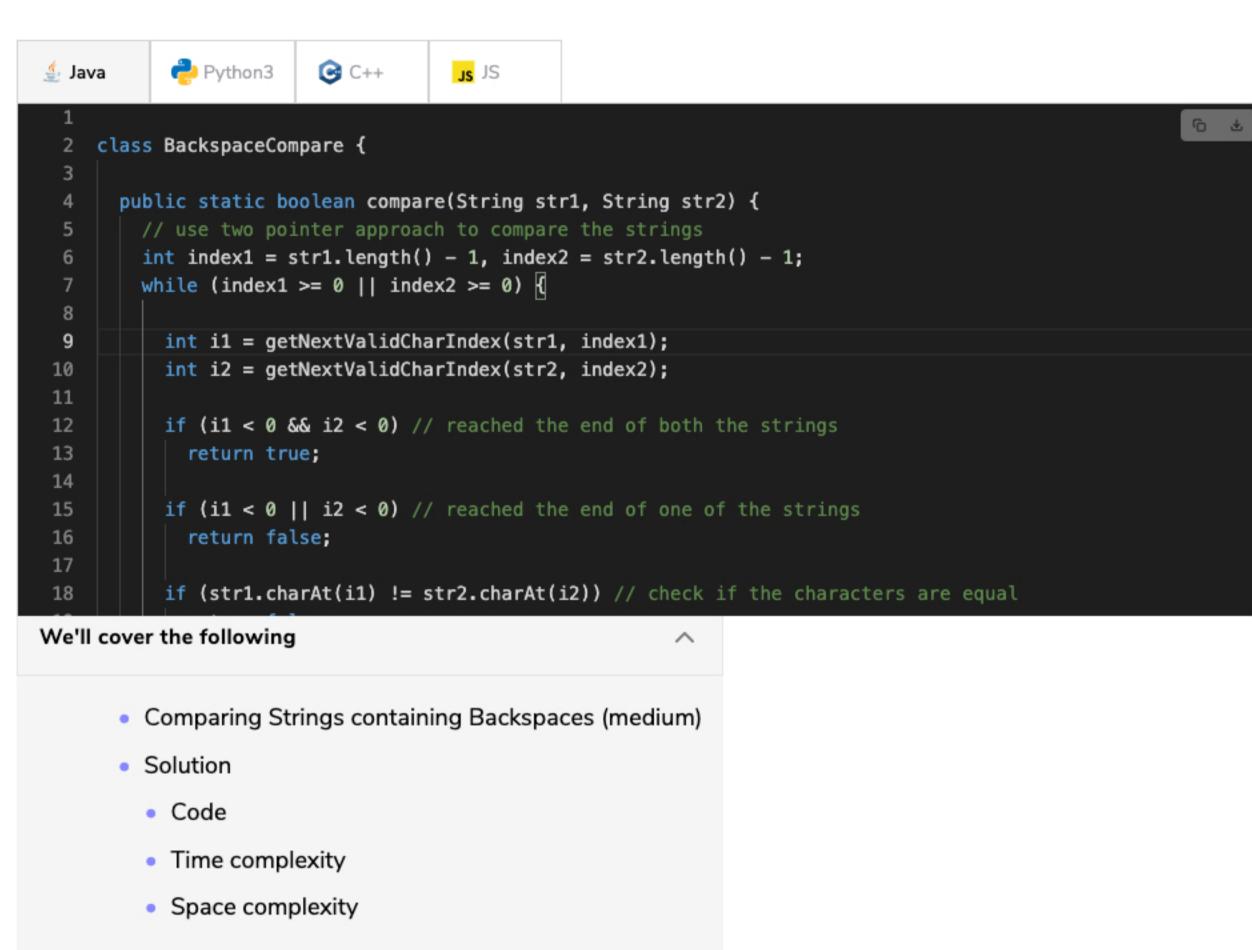
```
Input: str1="xywrrmp", str2="xywrrmu#p"
Output: true
Explanation: After applying backspaces the strings become "xywrrmp" and "xywrrmp" respectively.
```

#### Solution

To compare the given strings, first, we need to apply the backspaces. An efficient way to do this would be from the end of both the strings. We can have separate pointers, pointing to the last element of the given strings. We can start comparing the characters pointed out by both the pointers to see if the strings are equal. If, at any stage, the character pointed out by any of the pointers is a backspace ('#'), we will skip and apply the backspace until we have a valid character available for comparison.

#### Code

Here is what our algorithm will look like:



## Comparing Strings containing Backspaces (medium)

Given two strings containing backspaces (identified by the character '#'), check if the two strings are equal.

## Example 1:

```
Input: str1="xy#z", str2="xzz#"
Output: true
Explanation: After applying backspaces the strings become "xz" and "xz" respectively.
Example 2:
```

# Explanation: After applying backspaces the strings become "xz" and "xy" respectively. Example 3:

Output: false

Input: str1="xy#z", str2="xyz#"

```
Input: str1="xp#", str2="xyz##"
Output: true
Explanation: After applying backspaces the strings become "x" and "x" respectively.
In "xyz##", the first '#' removes the character 'z' and the second '#' removes the character 'y'.
```

# Example 4:

```
Input: str1="xywrrmp", str2="xywrrmu#p"
Output: true
Explanation: After applying backspaces the strings become "xywrrmp" and "xywrrmp" respectively.
```

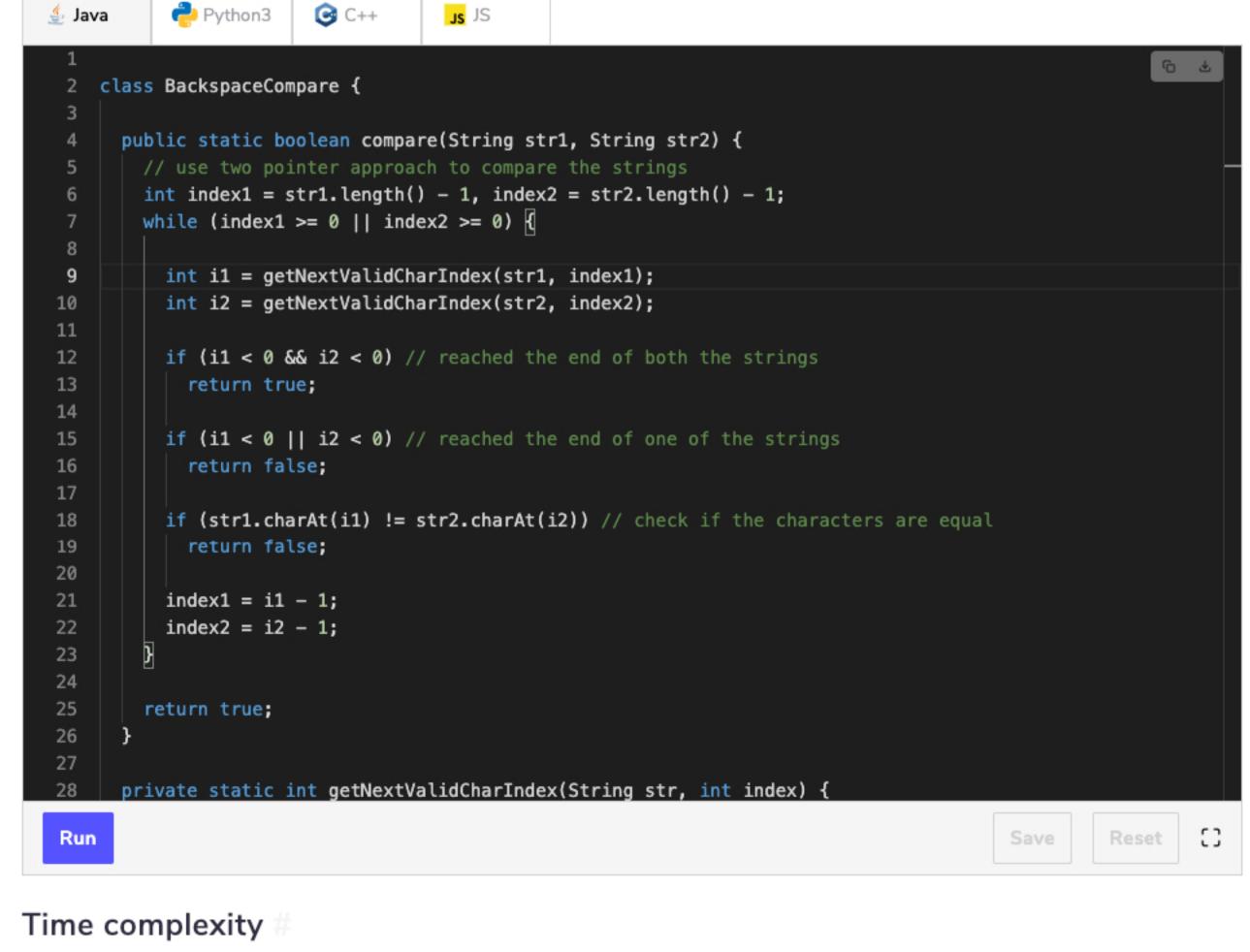
## To compare the given strings, first, we need to apply the backspaces. An efficient way to do this would be from the end of both the strings. We can have separate pointers, pointing to the last element of the given

Solution

from the end of both the strings. We can have separate pointers, pointing to the last element of the given strings. We can start comparing the characters pointed out by both the pointers to see if the strings are equal. If, at any stage, the character pointed out by any of the pointers is a backspace ('#'), we will skip and apply the backspace until we have a valid character available for comparison.

Code #

## Here is what our algorithm will look like:



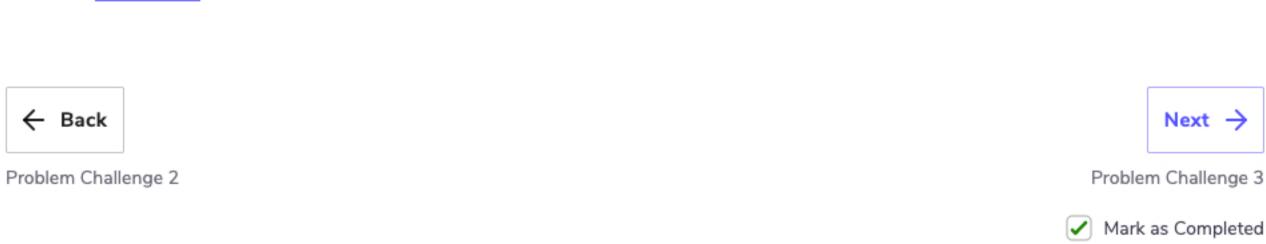
## The time complexity of the above algorithm will be O(M+N) where 'M' and 'N' are the lengths of the two

input strings respectively.

Space complexity #

## The algorithm runs in constant space O(1).

prep. See how ①



Want to work at Google, Facebook, or Amazon? Get hired faster with anonymous mock

interviews conducted by senior engineers from those companies. Detailed feedback helps you



 $\times$