



Master algorithms together on [Binary Search](#)! Create a room, invite your friends, and race to finish the problems.

Daily Coding Problem

[Blog](#)

Daily Coding Problem #159

Problem

This problem was asked by Google.

Given a string, return the first recurring character in it, or null if there is no recurring character.

For example, given the string "acbbac", return "b". Given the string "abcdef", return null.

Solution

We can solve this problem by keeping track of the characters we've seen in a set. Once we encounter a character we've seen already, we can return it.

```
def first_recurring(s):  
    seen = set()  
    for char in s:  
        if char in seen:  
            return char
```

```
        seen.add(char)
    return None
```

This runs in $O(n)$ time and space, since we have to iterate over the string.

We can improve the space complexity by using bitwise operators to set the bits of characters already seen in the string:

```
def first_recurring(str):
    checker = 0
    for c in str:
        val = ord(c) - ord('a')
        if (checker & (1 << val)) > 0:
            return c
        checker |= (1 << val)
    return None
```

Thanks to Shantanu for this solution.

© Daily Coding Problem 2019

[Privacy Policy](#)

[Terms of Service](#)

[Press](#)