Curso › Week 2... › 4. Func... › Exercis...

Audit Access Expires 5 de ago de 2020
You lose all access to this course, including your progress, on 5 de ago de 2020.

Exercise: is in

Finger Exercises due Aug 5, 2020 20:30 -03 *Completo*

Exercise: is in

5.0/5.0 points (graded)
ESTIMATED TIME TO COMPLETE: 18 minutes

We can use the idea of **bisection search** to determine if a character is in a string, so long as the string is sorted in alphabetical order.

First, test the middle character of a string against the character you're looking for (the "test character"). If they are the same, we are done - we've found the character we're looking for!

If they're not the same, check if the test character is "smaller" than the middle character. If so, we need only consider the lower half of the string; otherwise, we only consider the upper half of the string. (Note that you can compare characters using Python's `<` function.)

Implement the function `isIn(char, aStr)` which implements the above idea recursively to test if `char` is in `aStr`. `char` will be a single character and `aStr` will be a string that is in alphabetical order. The function should return a boolean value.

As you design the function, think very carefully about what the base cases should be.

```
def isIn(char, aStr):
    '''
    char: a single character
    aStr: an alphabetized string

    returns: True if char is in aStr; False otherwise
    '''
    # Your code here
    indice = len(aStr)//2
    if indice == 0:
        return False
    else:
        if aStr[indice] == char:
            return True
        else:
```

Press ESC then TAB or click outside of the code editor to exit

Correta

```
def isIn(char, aStr):
   '''
   char: a single character
   aStr: an alphabetized string

   returns: True if char is in aStr; False otherwise
   '''
   # Base case: If aStr is empty, we did not find the char.
   if aStr == '':
      return False

   # Base case: if aStr is of length 1, just see if the chars are equal
   if len(aStr) == 1:
      return aStr == char

   # Base case: See if the character in the middle of aStr equals the
   #   test character
   midIndex = len(aStr)//2
   midChar = aStr[midIndex]
   if char == midChar:
      # We found the character!
      return True

   # Recursive case: If the test character is smaller than the middle
   #  character, recursively search on the first half of aStr
   elif char < midChar:
      return isIn(char, aStr[:midIndex])

   # Otherwise the test character is larger than the middle character,
   #  so recursively search on the last half of aStr
   else:
      return isIn(char, aStr[midIndex+1:])
```

Test results

Hide output

CORRECT

```
Test: isIn('a', '')
```

Output:

```
False
```

```
Test: isIn('t', 'bcddhlotuxxz')
```

Output:

```
True
```

```
Test: isIn('p', 'bcemnppswx')
```

Output:

```
True
```

```
Test: isIn('z', 'eghhhuxz')
```

Output:

```
True
```

```
Test: isIn('x', 'adloppqtwzz')
```

Output:

```
False
```

```
Test: isIn('v', 'acchijjoppqrrx')
```

Output:

```
False
```

```
Test: isIn('u', 'cdginopquvw')
```

Output:

```
True
```

```
Test: isIn('t', 'fimoppz')
```

Output:

```
False
```

```
Test: isIn('n', 'dgiklmmnnsx')
```

Output:

```
True
```

```
Test: isIn('d', 'u')
```

Output:

```
False
```

Hide output

Note: In programming there are many ways to solve a problem. For your code to check correctly here, though, you must write your recursive function such that you make a recursive call directly to the function `isIn`. Thank you for understanding.

Hints

Basic function structuring

Be very careful about how you slice the string in the recursive cases! Before you execute the recursive cases, you test the middle character - so if you reach the recursive cases, you know the middle character cannot be a match, right? So be careful to *not* include this character when you make your recursive call!

What should your base case be?

You should be thinking about 3 situations:

- What happens when the string is empty?
- What happens when the string is of length 1?
- What happens when the test character equals the middle character?

What should your recursive case be?

You should be thinking about 2 situations:

- What happens when the test character is smaller than the middle character?
- What happens when it is larger?

If you are getting the error stating that "Your code should be recursive" when you already make a call to `isIn` : check your indention -- specifically, a common mistake is that your function and docstring do not start at the same indentation level.

Enviar

Answers are displayed within the problem

Exercise: is in

Ocultar discussão

Topic: Lecture 4 / Exercise: is in

Show all posts   por atividade recente

? I don't know where to start
I don't have any idea of where to start. I don't know if I'm difficulting the problem or not, but I do... 1

Syntax mistake with else?
Hi, it keeps telling me Syntax mistake with else in the second to last line. Can someone help? def i... 2

Something quite like an infinite loop 5

Hi, this code works well for most of the cases but sometimes it says : RecursionError: maximum r...

💬 Spoiler How Many Lines did it Take You?
I was just curious to know how many lines it took everyone to accomplish this task? I feel like min...
38

? (SPOILER) No output
Hi, I wrote the following code which returns a boolean if the original string (aStr) is one of the bas...
4

? [Contains Code] Anyone willing to help me figure out where I went wrong?
Here is my code. def isIn(char, aStr): ''' char: a single character aStr: an alphabetized string returns...
3

? SPOILER - My Code is included
So when I check this in Spyder, and in pythontutor.com, I get the expected results back - True or F...
3

? Why return when calling the function
In previous examples of recursion we do not return the function in for it to be used again recursiv...
2

💬 I don't even understand what this is asking
I guess I'm getting stuck on what this question is even asking? Is it "Determine if a certain charact...
2

💬 SPOILER! Want some feedback on my code if you please
Hi all! That one was challenging fun. I just wanted to get some feedback on my code. I added two ...
1

💬 (SPOILER) Sometimes works, other times an infinite loop
Hi! My code sometimes works (so I got a 3/5 so far) for inputs like **isIn('g', 'abeefgkklllmnppsvvw...
6

💬 frustration
i have been sitting for the past 2 hours trying to fix my code if aStr == '': return False if len(aStr)==...
4

💬 Base Case - Why Empty String?
Hi. I don't understand why we need to include the empty string base case. Through recursion, we ...
2