Problem 459 : Repeated Substring Pattern

<https://leetcode.com/problems/repeated-substring-pattern/>

Solution:

1. Get the length of the string and set it to n.

2. Since we need to check if the pattern is repeating, we only have to check half the length of the array.

3. We iterate through the first half of the array from index 1, and check if the first character is repeated.

4. If so, get the index of the character when it repeats, and see if it can divide n evenly (without remainder).

5. If so, the number of times the pattern will repeat will be (n / index of the repeated character).

6. Compare the given string with substring repeated (n/index of the repeated character). If they are equal return True.

7. If we iterate through the array and have not returned anything, return False.

-------------

class Solution:

def repeatedSubstringPattern(self, s: str) -> bool:

n = len(s)

for i in range(1, n//2 + 1):

if s[i] == s[0]:

if n % i == 0:

if s == s[0:i] \* int(n//i):

return(True)

return(False)