Two Strings ☆

Editorial by AllisonP

There are two concepts involved in solving this challenge:

1. Understanding that a single character is a valid substring.
2. Deducing that we only need to know that the two strings have a common substring — we don't need to know what that substring is.

Thus, the key to solving this challenge is determining whether or not the two strings share a common character because if they have a common character then they have a common substring of lengh 1.

To do this, we create two sets, a and b, where each set contains the unique characters that appear in the string it's named after. Because sets don't store duplicate values, we know that the size of our sets will never exceed the 26 letters of the English alphabet. In addition, the small size of these sets makes finding the intersection very quick.

If the intersection of the two sets is empty, we print `NO` on a new line; if the intersection of the two sets is not empty, then we know that strings a and b share one or more common characters and we print `YES` on a new line.

```
# Complete the twoStrings function below.
def twoStrings(s1, s2):
    # create sets of unique characters
    # and test for intersection
    if set(s1)&set(s2):
        return "YES"
    else:
        return "NO"
```

Tested by AllisonP

Problem Tester's code:

```
import java.util.*;

public class Solution {
    static Set<Character> a;
    static Set<Character> b;

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int n = scan.nextInt();
        for(int i = 0; i < n; i++) {
            a = new HashSet<Character>();
            b = new HashSet<Character>();

            //creating the set of string1
            for(char c : scan.next().toCharArray()) {
                a.add(c);
            }
            //creating the set of string2
            for(char c : scan.next().toCharArray()) {
                b.add(c);
            }

            // store the set intersection in set 'a'
            a.retainAll(b);

            System.out.println( (a.isEmpty()) ? "NO" : "YES" );
        }
        scan.close();
    }
}
```

STATISTICS

Difficulty:	Easy
Time Complexity:	$O(n)$
Required Knowledge:	Set theory, Strings, Characters
Publish Date:	Nov 14 2014

Originally featured in 101 Hack November'14

NEED HELP?

View discussions

View top submissions

Feedback

Was this editorial helpful?

Yes No