646. First Position Unique Character

* [Description](http://lintcode.com/en/problem/first-position-unique-character/#description)
* [Notes](http://lintcode.com/en/problem/first-position-unique-character/#note)
* [Testcase](http://lintcode.com/en/problem/first-position-unique-character/#testcase)
* [Judge](http://lintcode.com/en/problem/first-position-unique-character/#judge)

Given a string, find the first non-repeating character in it and return it's index. If it doesn't exist, return -1.

Have you met this question in a real interview?

Yes

**Example**

Given s = "lintcode", return 0.

Given s = "lovelintcode", return 2.

[http://lintcode.com/en/problem/first-position-unique-character/#](http://lintcode.com/en/problem/first-position-unique-character/)

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication5;

**import** java.util.Arrays;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.LinkedHashMap;

**import** java.util.Map;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication5 {

**public** **static** **int** firstUniqChar(String s) {

*// write your code here*

        LinkedHashMap<Character, Integer> hm =

**new** LinkedHashMap();

**for**(**int** i =0; i<s.length(); i++) {

**char** actual = s.charAt(i);

**if**(hm.containsKey(actual)) {

                hm.put(actual, hm.get(actual)+1);

            } **else** {

                hm.put(actual, 1);

            }

        }

**for**(**int** i =0; i<s.length(); i++) {

**if**(hm.get(s.charAt(i))==1) {

**return** i;

           }

       }

**return** -1;

    }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

        System.out.println(firstUniqChar( "lovelintcode" ));

    }

}

*---------------------OTRA FORMA --------------------------*

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication5;

**import** java.util.Arrays;

**import** java.util.HashMap;

**import** java.util.Iterator;

**import** java.util.LinkedHashMap;

**import** java.util.Map;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication5 {

***/\*\****

***\* @param args the command line arguments***

***\*/***

*/\* public static int firstUniqChar(String s) {*

*// write your code here*

*LinkedHashMap<Character, Integer> hm =*

*new LinkedHashMap();*

*for(int i =0; i<s.length(); i++) {*

*char actual = s.charAt(i);*

*if(hm.containsKey(actual)) {*

*hm.put(actual, hm.get(actual)+1);*

*} else {*

*hm.put(actual, 1);*

*}*

*}*

*int indice =0;*

*for(Character key : hm.keySet()) {*

*Integer val = hm.get(key);*

*if(val < 2) {*

*return indice;*

*}*

*indice++;*

*}*

*return -1;*

*}\*/*

**public** **static** String sortString(String inputString)

    {

*// convert input string to char array*

**char** tempArray[] = inputString.toCharArray();

*// sort tempArray*

        Arrays.sort(tempArray);

*// return new sorted string*

**return** **new** String(tempArray);

    }

**public** **static** **int** firstUniqChar(String s) {

*// write your code here*

        String ordenado =  sortString(s);

**for**(**int** i =0; i<s.length();) {

**char** actual = s.charAt(i);

**int** cont =0;

**int** indice = ordenado.indexOf(actual);

**while**(indice<ordenado.length() && ordenado.charAt(indice)== actual ) {

                cont++;

                indice++;

**if**(cont > 1) **break**;

            }

**if**(cont == 1) {

**return** s.indexOf(actual);

            }

            i++;

        }

**return** -1;

    }

**public** **static** **void** main(String[] args) {

*// TODO code application logic here*

        System.out.println(firstUniqChar( "lovelintcode" ));

    }

}