891. Valid Palindrome II

* [Description](http://www.lintcode.com/en/problem/valid-palindrome-ii/" \l "description)
* [Notes](http://www.lintcode.com/en/problem/valid-palindrome-ii/#note)
* [Testcase](http://www.lintcode.com/en/problem/valid-palindrome-ii/#testcase)
* [Judge](http://www.lintcode.com/en/problem/valid-palindrome-ii/#judge)

Given a non-empty string s, you may delete at most one character. Judge whether you can make it a palindrome.

 Notice

The string will only contain lowercase characters a-z. The maximum length of the string is 50000.

Have you met this question in a real interview?

Yes

**Example**

Given s = "aba"return true  
Given s = "abca"return true // delete c

<http://www.lintcode.com/en/problem/valid-palindrome-ii/#>

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

***\*/***

**public** **static** **boolean** validPalindrome(String s) {

*// Write your code here*

        HashMap <Character, Integer> hm=

**new** HashMap();

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

**if**(hm.containsKey(s.charAt(i))) {

                hm.put(s.charAt(i), hm.get(s.charAt(i))+1);

            } **else** {

                 hm.put(s.charAt(i), 1);

            }

        }

**int** impares =0;

**int** unos =0;

**for**(**char** ch : hm.keySet()) {

**if**(hm.get(ch) % 2 != 0 && hm.get(ch) != 1) {

                impares++;

            }

**if**(hm.get(ch) == 1) {

                unos++;

            }

        }

**return** (unos <= 2 && impares < 2 );

    }

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

*// TODO code application logic here*

*//String s = "ognfjhgbjhzkqhzadmgqbwqsktzqwjexqvzjsopolnmvnymbbzoofzbbmynvmnloposjzvqxejwqztksqwbqgmdazhqkzhjbghjfno";*

       String s = "abca";

       System.out.println(validPalindrome(s));

    }

}