**[Valid Palindrome II](https://leetcode.com/problems/valid-palindrome-ii/)**

**public** **class** ValidPalindromeII {

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

// **TODO** Auto-generated method stub

System.***out***.println(*validPalindrome*("abc"));

}

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

**if**(s == **null** || s.length() <= 2) {

**return** **true**;

}

**char**[] map = s.toCharArray();

**for**(**int** i = 0 , j = s.length() - 1 ; i < s.length() && i < j ; i++, j--) {

**if**(map[i] != map[j]) {

**return** *isValid*(map , i + 1 , j) || *isValid*(map , i , j - 1);

}

}

**return** **true**;

}

**public** **static** **boolean** isValid(**char**[] map , **int** i , **int** j) {

**while**(i <= j) {

**if**(map[i] != map[j]) {

**return** **false**;

}

i++;

j--;

}

**return** **true**;

}

}

Time Complexity : O(n) , n is length is given string

Space Complexity : O(n), n is length is given string