**Given a string s, return true *if the* s *can be palindrome after deleting at most one character from it*.**

**Example 1:**

**Input: s = "aba"**
**Output: true**

**Example 2:**

**Input: s = "abca"**
**Output: true**
**Explanation: You could delete the character 'c'.**

**Example 3:**

**Input: s = "abc"**
**Output: false**

**Constraints:**

- **1 <= s.length <= $10^5$**
- **s consists of lowercase English letters.**

## Solution:

```
class Solution {
    public boolean validPalindrome(String s) {
        int left = 0, right = s.length() - 1;
        while (left < right) {
            if (s.charAt(left) != s.charAt(right)) {
                return isPalindrome(s, left + 1, right) || isPalindrome(s, left, right - 1);
            }
            left++;
            right--;
        }
        return true;
    }

    private boolean isPalindrome(String s, int left, int right) {
        while (left < right) {
            if (s.charAt(left) != s.charAt(right)) {
                return false;
            }
            left++;
            right--;
        }
        return true;
    }
}
```