<https://leetcode.com/problems/valid-palindrome-ii/>

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 <= 105
* s consists of lowercase English letters.

**Attempt 1: 2023-02-25**

**Solution 1: Two Pointers (10 min)**

class Solution {

public boolean validPalindrome(String s) {

int i = 0;

int j = s.length() - 1;

while(i < j) {

if(s.charAt(i) == s.charAt(j)) {

i++;

j--;

} else {

return isPalindrome(s, i + 1, j) || isPalindrome(s, i, j - 1);

}

}

return true;

}

private boolean isPalindrome(String s, int i, int j) {

while(i < j) {

if(s.charAt(i) != s.charAt(j)) {

return false;

}

i++;

j--;

}

return true;

}

}

**Refer to**

<https://leetcode.com/problems/valid-palindrome-ii/solutions/1904942/java-c-awesome-visuall-explanation/>

Let's consider on example in order to understand it :-**Input**: s = "abcdecba"**Output**: true

So, what I can do is create two pointer's & start comparing them from.

One will start from 0th Index & another will start from last index.

We'll check, if they are equal then continue checking

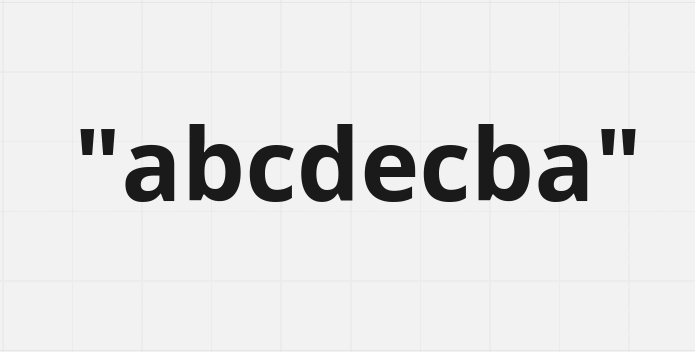
But if they are undequal we can have 2 cases :-

Case 1 : Skip e to check whether it's forming an palindrome

Case 2 : Skip d to check whether it's forming an plaindrome

But still if after deleting one character we are not getting palindrome **return false**

Otherwise **return true**



class Solution {

public boolean validPalindrome(String s) {

int i = 0;

int j = s.length() - 1;

while(i <= j){

if(s.charAt(i) == s.charAt(j)){

i++;

j--;

}

else return isPalindrome(s, i + 1, j) || isPalindrome(s, i, j - 1);

}

return true;

}

public boolean isPalindrome(String s, int i, int j){

while(i <= j){

if(s.charAt(i) == s.charAt(j)){

i++;

j--;

}

else return false;

}

return true;

}

}