


Valid Palindrome 2

686

IP \rightarrow String \rightarrow at most 0/1 removed

↓
Palindrome $\begin{cases} \text{true} \\ \text{false} \end{cases}$

for ex

NOON
SAAS
RACECAR
MADAM
BABAB

left \rightarrow right
right \rightarrow left } some

aba \rightarrow 0 removed

\hookrightarrow Palindrome \rightarrow true

ab~~x~~a \rightarrow aba \rightarrow 1 removed

abc \rightarrow X

↓
Palindrome

↓
ab
bc

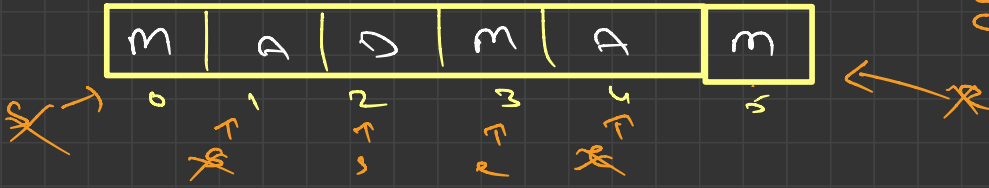
Approach

reverse string \rightarrow then match

2 ways

only one removal allowed

equal
str
c--



if not equal \rightarrow remove

\downarrow
remove

\downarrow
remaining \rightarrow palindrome

we can do one thing

remove 'D' & save one

remove 'M' & save one

} in any case
the result
are palindrome
 \downarrow
true

M A D A M

A D M A M

} valid
palindrome

m	A	D	A	R	R	A	C	A	m
---	---	---	---	---	---	---	---	---	---

\uparrow^0 \uparrow^1 \uparrow^2 \uparrow^3 \uparrow^4 \uparrow^5 \uparrow^6 \uparrow^7 \uparrow^8 \uparrow^9

$s[0] == s[9] \rightarrow \text{true} \rightarrow i++, j--$

$s[1] == s[8] \rightarrow \text{true} \rightarrow i++, j--$

$s[2] == s[7] \rightarrow \text{false}$

\downarrow
 1 remove $\rightarrow ?$
 D remove C remove
 A R R A C D A R R A
 Not a palindrome

return false

~~| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | B | C | R | R | R | C | A |
|---|---|---|---|---|---|---|---|~~

~~0~~ ~~1~~ ~~2~~ 3 4 5 6 ~~7~~
~~---~~ ~~---~~ ~~---~~ ~~---~~

$s[0] == s[7] \rightarrow \text{true} \quad \text{---}, \text{---}$

$s[1] == s[6] \rightarrow \text{false}$

↓
1 removed

~~B → or + C~~

↓
C R R R C

↓
Palindrome

B C R R R

↓
not Palindrome

↓
final
result

if any of the one
is palindrome

the result is
true

bool checkPalindrome (string s, int i, int j)

```
{
    while ( i < j )
    {
        if ( s[i] != s[j] )
            return false;
        else
            i++, j--;
    }
    return true;
}
```

bool validPalindrome (string s)

```
int i = 0;
int j = s.length() - 1;
```

```
while ( i < j )
{
    if ( s[i] == s[j] )
    {
        i++, j--;
    }
    else
    {
        // s[i] != s[j] → 1 removal allowed

```

// check palindrome for remaining string after removal

// remove ith char

```
bool ans1 = checkPalindrome ( s, i+1, j );
```

// remove jth char

```
bool ans2 = checkPalindrome ( s, i, j+1 );
```

```
    return ans1 || ans2;
}
```

// if here → 0 removal → valid palindrome
return true;

