

Dry run of Plug-in problem

eq \rightarrow	$s =$	0 1 2 3 4 5 6 7 8 9 10 11 12 13 14	$s[i]$	$s[i+1]$	same / not	str 2	
0	a	a b a c a b a a b a c a b a a	b	a	x	a	
1	b	a b a c a b a a b a c a b a a	a	b	x	ab	
2	a	a b a c a b a a b a c a b a a	c	a	x	aba	
3	c	a b a c a b a a b a c a b a a	a	c	x	abac	
4	a	a b a c a b a a b a c a b a a	b	b	x	abaca	New changed
5	b	a b a c a b a a b a c a b a a	a	a	x	abacab	/ become true
6	a	a b a c a b a a b a c a b a a	a	a	✓	abacab	
7	b	a b a c a b a a b a c a b a a	a	b	x	abacabb	
8	a	a b a c a b a a b a c a b a a	c	a	x	abacabba	
9	c	a b a c a b a a b a c a b a a	a	c	x	abacabba	
10	a	a b a c a b a a b a c a b a a	b	a	x	abacabbac	
11	b	a b a c a b a a b a c a b a a	a	b	x	abacabbaca	
12	a	a b a c a b a a b a c a b a a	b	a	x	abacabbacab	
13	a	a b a c a b a a b a c a b a a	a	a	✓	abacabbacab	

New $s = abacabbacab$

0 1 2 3 4 5 6 7 8 9 10
a b a c a b b a c a b

Again continue same process.

keep will always work until we have satisfied the inside break condition.

Dry run for stack approach

eq \rightarrow meallazy

current char	ans before	Action	ans after
m	" "	push	m
e	m	push	me
a	me	push	mea
l	mea	push	meal
l	meal	pop	mea
a	mea	pop	me
z	me	push	mez
y	mez	push	mezy

```

for (char c : s) {
    if (!ans.empty() & ans.back() == c) {
        ans.pop_back();
    } else {
        ans.push_back(c);
    }
}

```

This means if ans is not empty & current character c is same as last character in ans, then remove last character from ans.

else,

If the stack is empty OR
current character is different, keep the character.

- * Each character is pushed once, can be popped once.
so Time complexity = $O(n)$.