## 214. Shortest Palindrome

Attempted

Hard | Topics | Companies

You are given a string `s`. You can convert `s` to a palindrome by adding characters in front of it.

Return *the shortest palindrome you can find by performing this transformation*.

**Example 1:**

```
Input: s = "aacecaaa"
Output: "aaacecaaa"
```

**Example 2:**

```
Input: s = "abcd"
Output: "dcbabcd"
```

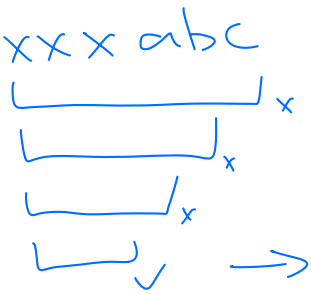
shortest to add

↳ longest palidrome start with index 0 existed

+ reverse the remaining part.

```python
def helper_pali(s, l, r):
    while l <= r:
        if s[l] != s[r]:
            return False
        l += 1
        r -= 1
    return True
```

Eg. xxxabc



→ cbaxxxabc

```python
class Solution:
    def shortestPalindrome(self, s: str) -> str:
        def helper_pali(s, l, r):
            while l <= r:
                if s[l] != s[r]:
                    return False
                l += 1
                r -= 1
            return True

        for i in reversed(range(len(s))):
            if helper_pali(s, 0, i):
                suf = s[i+1: ]
                return suf[::-1] + s

        return ""
```

Ln 11, Col 41 | Saved | Run

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Time Limit Exceeded

↳ kmp

Rabin karp ✓

Rabin karp

turn string comparison to integer comparison

```python
class Solution:
    def shortestPalindrome(self, s: str) -> str:
        prefix = 0
        suffix = 0
        base = 29
        last_index = 0 # -1
        power = 1
        mod = 10**9 + 7

        for i, c in enumerate(s):
            char = (ord(c) - ord('a') + 1)

            prefix = (prefix * base) % mod
            prefix = (prefix + char) % mod
            suffix = (suffix + char * power) % mod
            power = (power * base) % mod

            if prefix == suffix:
                last_index = i
        suffix = s[last_index + 1:]
        return suffix[::-1] + s
```

Eg. zzzabc  a→1, z→26

base=29 prime number, less collision

| i | prefix | | suffix | match | last_index |
|---|---|---|---|---|---|
| 0 | z→26 | = | z→26 | ✓ | 0 |
| 1 | zz→26×29¹+26 | = | zz→26+26×29¹ | ✓ | 1 |
| 2 | zzz→26×29²+26×29¹+26 | = | zzz→26×29²+26×29¹+26 | ✓ | 2 |
| 3 | zzza→26×29³+26×29²+26×29+1 | ≠ | azzz→1×29³+26×29²+26×29+26 | ✗ | 2 |
| 4 | | | | ✗ | 2 |
| 5 | | | | ✗ | 2 |

→ last index = 2

remaining = s[2+1:]

return remaining[::-1] + s

to avoid overflow, mod a very large prime number say, $10^7+9$

Runtime: 27ms

Python

```python
class Solution:
    def shortestPalindrome(self, s: str) -> str:
        i, n=0, len(s)
        for c in s[::-1]:
            if c==s[i]:i+=1
        if i==n: return s
        sub=s[i:]
        return sub[::-1]+self.shortestPalindrome(s[0:i])+sub
```

This is the best solution so far.

Jan 7th, 2025