

Longest Happy Prefix

Prefix = suffix — directly the KMP LPS ■

● HARD

■ Hook

Find the longest prefix of string s that is also a suffix (but not the whole string). This IS literally the KMP LPS array's last value. Once you understand KMP, this problem is trivial.

■ Core Intuition

$\text{lps}[n-1]$ = longest proper prefix of s that is also a suffix. Return $s[0..\text{lps}[n-1]-1]$. If $\text{lps}[n-1]=0$, return empty string.

■ Visual — $s = \text{'levelevel'}$

```
s = 'levelevel'
lps = [0,0,0,1,2,3,4,5,0] ← wait, let's compute properly
s = 'ababab'
lps = [0,0,1,2,3,4]
lps[5] = 4 → longest prefix=suffix = 'abab'
s = 'aabaa'
lps = [0,1,0,1,2] → lps[4]=2 → answer = 'aa'
```

■ Code

```
public String longestPrefix(String s) {
    int n = s.length();
    int[] lps = new int[n];
    int len=0, i=1;
    while (i < n) {
        if (s.charAt(i) == s.charAt(len))
            lps[i++] = ++len;
        else if (len != 0)
            len = lps[len-1];
        else
            lps[i++] = 0;
    }
    return s.substring(0, lps[n-1]);
}
```

■ TIME

$O(n)$

■ SPACE

$O(n)$

- ✓ Directly apply KMP LPS array
- ✓ $\text{lps}[n-1]$ = length of longest proper prefix = suffix
- ✓ Return `s.substring(0, lps[n-1])`
- ✓ If $\text{lps}[n-1]=0 \rightarrow$ no happy prefix $\rightarrow$ return ""
- ✓ Time: $O(n)$ | Space: $O(n)$

■ MENTAL MODEL — Remember This Forever

This is the purest application of the KMP LPS array. The LPS array was BUILT to track prefix-suffix matches. $\text{lps}[i]$ literally IS the answer to 'what is the longest string that is both a prefix and suffix of $s[0..i]$ '?