Program Name- Longest Common Extension / LCE | Set 1 (Introduction and Naive Method)

Project Category- Strings

What is Longest Common Extension / LCE ?-

We start by explaining the term – "Longest Common Extension / LCE".

Longest Common Extension Problem considers a string- str and several queries of the form- (L, R). In each of the query we have to answer the length of the longest common prefix starting at index- L and R.

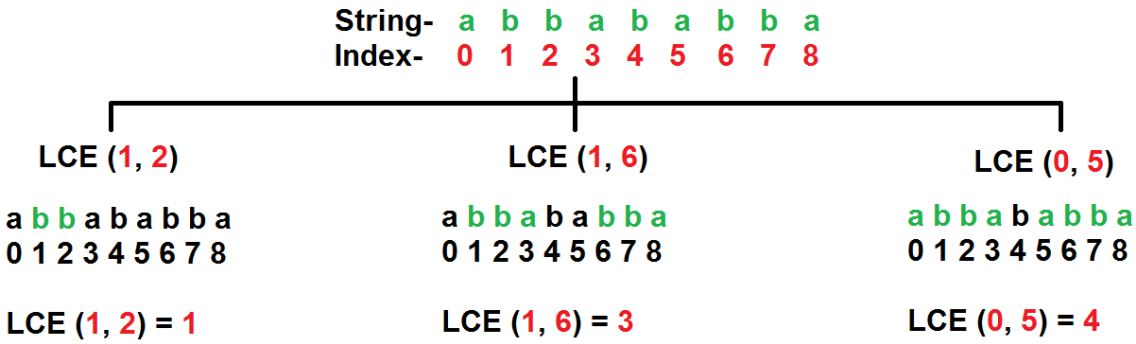
Example-

For example, we consider a string – "abbababba". We have three queries to answer- LCE(1, 2) , LCE(1, 6) and LCE(0, 5).

We have to find the length of the longest common prefix starting at index- (1, 2), (1, 6) and (0, 5).

Consider the below illustration-

The string highlighted "green" are the longest common prefix starting at index- L and R of the respective queries.



Algorithm (Naive Method)-

1) For each of the LCE queries of the form – LCE(L, R) do the following:-

a) Initialise the LCE 'length' as 0
b) Start comparing the prefix starting from index- L and R character by character.
c) If the characters matches, then this character is in our Longest Common Extension. So increment 'length' (length++).
d) Else if the characters mismatch, then return this 'length'.

2) The returned 'length' will be the required LCE(L, R).

In the next sets we will discuss how LCE (Longest Common Extension) problem can be reduced to a RMQ (Range Minimum Query). We will also discuss more efficient methods to find the longest common extension.

Time Complexity-

The time complexity is O(Q.N), where
Q = Number of LCE Queries
N = Length of the input string

Auxiliary Space -

No auxiliary space is needed. This is an O(1) in-place algorithm.

An Interesting Fact-

One may be surprised that the although having a greater asymptotic time complexity, the naive method outperforms other efficient method(asymptotically) in practical uses.

The performance of the naive method will be discussed in more detail in Set 3.

Applications-

The LCE problem has many applications. Some of them are listed below-

1) K-Mismatch Problem → Landau-Vishkin uses LCE as a subroutine to solve k-mismatch problem
2) Approximate String Searching.
3) Palindrome Matching with Wildcards.
4) K-Difference Global Alignment.

References-

http://www.sciencedirect.com/science/article/pii/S1570866710000377