

1.

When developing my program, my strings initially included only strings provided in Chapter 15 as well as strings used to explain how the longest common algorithm worked (Geeks4geeks, tutorialspoint, etc). The goal was to find verifiable input so I can be sure my output is correct. Once I was sure I had the correct implementation, I verified it using strings that make up DNA and genomes provided by various biology websites. This was interesting to see how our LCS problem in Computer Science is actually a very invaluable algorithm used in genome sequencing.

2.

The runtime of PRINT_LCS and LCS_LENGTH are the same for both the dynamic and memoization implementation. PRINT_LCS is $O(m + n)$ since both implementations perform

constant time checks to decrement either i or j .

LCS_LENGTH is $O(mn)$ for both implementations since it builds the same table. The memoization and dynamic approach are separated by the way in which they are built

(e.g. top-down vs bottom-up).

Thus, both implementations have runtimes of $T(n) = mn + m + n$. It is important to note that dynamic implementation is much a simpler implementation than memoization.

Furthermore, both implementations drastically outperform an equivalent brute force approach which has an upper bound of $O(2^n)$.