

DA511H: Data Structures and Algorithms

Assignment – I

Submission Date: 12-Sep-2025

Dynamic Programming

- Given two sequences $X = \langle x_1, x_2, \dots, x_m \rangle$ and $Y = \langle y_1, y_2, \dots, y_n \rangle$, find a maximum length common subsequence (LCS) of X and Y . The LCS is a sequence that appears in both X and Y (in the same order), but not necessarily consecutively, and is the longest such sequence.

$$\begin{array}{c} X = \langle A, B, C, B, D, A, B \rangle \\ Y = \langle B, D, C, A, B, A \rangle \end{array}$$

The longest sequence is $\langle B, C, B, A \rangle$.

Hashing

11.2-1

Suppose we use a hash function h to hash n distinct keys into an array T of length m . Assuming simple uniform hashing, what is the expected number of collisions? More precisely, what is the expected cardinality of $\{|k, l| : k \neq l \text{ and } h(k) = h(l)\}$?

11.2-2

Demonstrate what happens when we insert the keys 5, 28, 19, 15, 20, 33, 12, 17, 10 into a hash table with collisions resolved by chaining. Let the table have 9 slots, and let the hash function be $h(k) = k \bmod 9$.

11.3-1

Suppose we wish to search a linked list of length n , where each element contains a key k along with a hash value $h(k)$. Each key is a long character string. How might we take advantage of the hash values when searching the list for an element with a given key?

11.3-2

Suppose that we hash a string of r characters into m slots by treating it as a radix-128 number and then using the division method. We can easily represent the number m as a 32-bit computer word, but the string of r characters, treated as a radix-128 number, takes many words. How can we apply the division method to compute the hash value of the character string without using more than a constant number of words of storage outside the string itself?