University of New South Wales (UNSW)

Algorithm Design & Analysis

Formatif Submission

(R) Subsequence Counting

Submitted By:
Zachary Xia
z5447429
2024/11/08 21:37

Tutor:
Larissa Yip

November 8, 2024

(a) This problem is similar to 3.07 Subsequence Search. Briefly explain why a similar greedy approach will not work here.

(b) Design an algorithm that runs in $O(nm)$ time to count the number of occurrences of A as a subsequence of B.

(a) This question asks us to count all possible sequences in the target string. The greedy solution only looks at the first occurrence of each character, so it won't work.

(b)

Subproblem: We define P(i,j) to be find the count the number of occurrences of A[a1…aj] as a subsequence of B[b1…bj]. The maximum number of occurrences of P(i, j) is denoted by opt(i, j)

Recurrence: opt(i,j) = opt(i – 1, j) + opt(i – 1, j - 1) if i == j, opt(i , j - 1) if i != j

After we solve all (i, j), the final answer is opt(n, m)

base case: opt(0,j) = 1, since empty string is a substring of any string

order of computation, increase i first, then increase j

The time complexity is O(mn) since we solve the subproblem mn times in total and store the value in opt 2d array, and each subproblem has O(1) time complexity.

The algorithm is correct because we consider all possible cases to count the number of occurrences, when i == j, we count the case of both include A[i] and not include A[i], and when i != j, we count the number of occurrence from A[0…i] and B[0…j - 1] to avoid lose track of dp. So the final answer is also correct.