Subsequence Counting

This is a **regular task**. You must submit a PDF, which can be produced using the LaTeX template on Moodle, exported from a word processor, hand-written or any other method.

We say that a string A occurs as a subsequence of another string B if we can obtain A by deleting some of the letters of B.

Suppose that Alice has a favourite string $A = a_1 \dots a_n$ of length n and Bob has a favourite string $B = b_1 \dots b_m$ of length m. Alice argues that Bob copied her string but changed some of the letters in between. Alice is curious about the number of ways Bob could have done this. That is, Alice wants to count the number of times A appears as a subsequence B. For example, suppose

$$A = \texttt{dang}$$
$$B = \texttt{i am good at dynamic programming}$$

Then A occurs as a subsequence of B precisely 7 times:

- i am good at **d**yn**a**mic programmi**ng**
- i am good at **d**ynamic progr**a**mmi**ng**
- i am goo**d** **a**t dy**n**amic pro**g**ramming
- i am goo**d** **a**t dy**n**amic programmin**g**
- i am goo**d** **a**t dynamic programmi**ng**
- i am goo**d** at dynamic progr**a**mmi**ng**
- i am goo**d** at dyn**a**mic programmi**ng**

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

Advice. Your solution should include:

- A clear subproblem definition.
- Base cases for your subproblem definition.
- A well-defined recurrence with respect to your subproblem definition and base cases.
- Some output that solves the original problem, as a function of the results generated by your recurrence.
- A correct order of computation, with respect to your recurrence.
- Time complexity analysis for your algorithm.
- Justification that your algorithm solves the problem correctly, with specific reference to the correctness of the base case(s), recurrence and overall answer.

Expected length: Up to a page.