

6 Algorithms I (FMS)

A *palindrome* is a string that, if reversed, remains the same, for example “madamimadam”. A *subsequence* of a string  $x$  is one obtained by dropping zero or more characters from  $x$  and taking the remaining ones in order: for example “tan” is a subsequence of “pentagon”. In this question you must find the longest palindrome subsequence (LPS) of a given string. [Note that the LPS may not be unique.]

- (a) Explain why it is possible to apply dynamic programming to the LPS problem. Develop and explain a recursive equation for the length of the LPS. [6 marks]
- (b) Develop and describe in detail, with pictures where appropriate, a bottom-up dynamic programming algorithm to solve the LPS problem. Include an explanation of how to recover the LPS from the bottom-up table you build. If you use pseudocode (not required), keep each pseudocode chunk under 10 lines and comment it clearly. Incomprehensible code will be scored as wrong. [9 marks]
- (c) Derive the asymptotic worst-case running time of your algorithm. [2 marks]
- (d) What else would you have to do to recover *all* the LPSs of a given string? [3 marks]