7 Algorithms (fms27)

Consider binary trees whose nodes have three fields: key (a single character between U and Z), left subtree, right subtree, with the two subtrees either both empty or both non-empty. Assuming suitable constructors, indicate an empty tree as $T()$ and a non-empty tree as $T(\text{key, leftSubtree, rightSubtree})$.

(a) Define unambiguous pre-order, in-order and post-order representations for such a tree $t$, called $r_{\text{pre}}(t)$, $r_{\text{in}}(t)$, $r_{\text{post}}(t)$, consisting of strings over the \{U, V, W, X, Y, Z, (, )\} alphabet, with balanced brackets, with three characters (two brackets and a letter) for each node, starting and finishing with a bracket unless $t$ is empty. Formally describe your three representations for a generic tree $t$, then produce the corresponding strings for the following tree. [6 marks]

(b) In this obfuscated pseudocode, the input $v0$ is a syntactically correct $r_{\text{post}}(t)$. Clearly explain (i) the purpose of the code; (ii) how it works; and (iii) how one should invoke it. Substitute meaningful explanatory identifiers for those $T_x$, $v_y$ and $m_z$. Identifiers $v4$ and $v6$ are worth more marks than the others. [7 marks]

(c) Explain in detail (i) how line 16 works, and (ii) why $v4$ will never be uninitialised when line 16 is executed. [4 marks]

(d) Write clear pseudocode that takes a Tree $t$ and returns $r_{\text{in}}(t)$. [3 marks]