Algorithms I

The product of two \( n \times n \) matrices, \( A \) and \( B \), is a third \( n \times n \) matrix, \( Z \), where

\[
Z_{ij} = \sum_{k=1}^{n} A_{ik} B_{kj}
\]

(a) A programmer directly implements this formula when writing a function to multiply two matrices. Find the asymptotic time complexity of such an algorithm, taking care to justify your answer. [3 marks]

(b) An alternative strategy to compute \( Z \) is to divide \( A \) and \( B \) into four \( \frac{n}{2} \times \frac{n}{2} \) matrices. Computing \( Z \) then involves eight \( \frac{n}{2} \times \frac{n}{2} \) matrix multiplications followed by a series of matrix additions. This approach is then applied recursively.

(i) Identify the algorithmic strategy in use. [1 mark]

(ii) Show that the run time of this alternative strategy is given by the recurrence relation

\[
t_n = K t_{[f(n)]} + O(g(n))
\]

where \( t_n \) is the time to compute the product of two \( n \times n \) matrices, \( K \) is a constant you should determine, and \( f(n) \) and \( g(n) \) are functions that you should identify. [6 marks]

(iii) Solve the recurrence relation to find an asymptotic complexity for \( t_n \). [5 marks]

(c) An optimisation of the algorithm presented in part (b) means that only seven matrix multiplications are needed rather than the eight previously suggested. State the new recurrence relation and solve it to show that this algorithm is \( O(n \log_2 7) \). [5 marks]