

科目：演算法 A

日期：106 年 1 月 19 日 第 1 頁 共 2 頁

請“✓”明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. 15% Consider the Strassen's matrix multiplication algorithm (a divide and conquer approach).

(1) Using traditional matrix product method, how many multiplications and additions do we need to multiply two 2×2 matrices? How many multiplications and additions do we need to multiply two 2×2 matrices for Strassen's method?

(2) Write down the related recurrence equation, and give the asymptotic tight bound solution. (No need to give any explanation)

(3) Suppose that we use a similar divide and conquer approach to compute this problem; divide $n \times n$ matrices into $\frac{n}{2} \times \frac{n}{2}$ submatrices the same way as Strassen's, while in the "merge" part, there are m submatrix multiplications and a submatrix additions. Find the time complexity of this method.

(4) Suppose that we divide the matrices into $\frac{n}{3} \times \frac{n}{3}$ submatrices (and there are m submatrix multiplications and a submatrix additions in the "merge" part). Find the time complexity of this method.

(5) For part (4), what is the largest value of m (submatrix multiplication) to get an asymptotic faster matrix multiplication algorithm than Strassen's?

2. 20% For each of the following three problems, decide whether it is a P-problem or an NP-hard problem. If it is P, name a polynomial time algorithm to solve it and justify your answer briefly. If it is NP-hard, show it.

- (1) Find a shortest simple path between two nodes in directed graph with negative weight directed cycle.
- (2) Find a negative weight cycle in a weighted directed graph.
- (3) Assume that there is NO negative weight directed cycles. Is the problem of finding a shortest simple path between two nodes P or NP-hard? Explain it briefly.

科目：演算法 A

日期：106 年 1 月 19 日 第 2 頁 共 2 頁

請“✓”明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

3. 15% Assume that $NP \neq P$. Classify whether each of the following problems belongs to P or NP-hard (or NP-complete). No need to explain. 1 point for each correct answer. For each wrong answer, deduct 1 point.

- (1) Find a shortest simple path between two nodes in a directed graph with positive edge weights.
- (2) Find a longest simple path between two nodes in a directed graph with positive edge weights.
- (3) Find a shortest simple path between two nodes in a directed graph with all the edges having negative edge weights.
- (4) Find a longest simple path between two nodes in a directed graph with all the edges having negative edge weights.
- (5) Find a smallest cycle in a graph, where all the edge-weights are one.
- (6) Find a largest cycle in a graph, where all the edge-weights are one.
- (7) 2-CNF Satisfiability problem.
- (8) Find ρ -approximation solution for the vertex-cover problem, for some ρ .
- (9) Find ρ -approximation solution for the traveling-salesman problem, for some ρ .
- (10) Find a minimum cut in a network..
- (11) Find a maximum cut in a network..
- (12) Find a maximum independent set in interval graphs.
- (13) Find a minimum vertex cover in interval graphs.
- (14) The traveling-salesman problem in graphs with triangle inequality.
- (15) Find ρ -approximation solution for the traveling-salesman problem in graphs with triangle inequality, for some ρ .

國立交通大學試題紙

一百零五學年度第一次
博士班資格考

科目：演算法 B

日期：106 年 1 月 19 日 第 1 頁 共 1 頁

請“✓”明 ✓不可看書 可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) Show how to sort N integers efficiently in the range 0 to $N^3 - 1$ in linear time.
2. (15%) Give a Huffman binary coding tree for the following data: *ABATDABEBAAAAAABEB*. Build it step by step. State the algorithm for building Huffman binary coding tree.
3. (10%) Give a dynamic programming algorithm for the 0-1 Knapsack problem.
4. (15%) Given two integer arrays $A[1..n]$ and $B[1..n]$, find the longest subsequence i_1, \dots, i_k , such that $A[i_1], A[i_2], \dots, A[i_k]$ are increasing and $B[i_1], B[i_2], \dots, B[i_k]$ are decreasing. Give an efficient algorithm and analyze its complexity.