

科目：演算法 A

日期：102 年 1 月 31 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. (12%) For the Table-Insert and Table-Delete algorithms given in the textbook, let $\Phi(T) = 2 * \text{num}[T] - \text{size}[T]$, if $\alpha \geq 1/2$; and $\text{size}[T]/2 - \text{num}[T]$, if $\alpha < 1/2$. Prove that the amortized time complexity for each Table-Delete is at most 3.
2. (8%) Student Imbest proposes a divide-and-conquer algorithm for minimum spanning trees, as follows.
Equally divide the set V of vertices into two sets V_1 and V_2 such that $||V_1| - |V_2|| \leq 1$. Let E_1 (E_2) be the edges induced from V_1 (V_2). Recursively solve the minimum-spanning-tree problems on both graphs, (V_1, E_1) and (V_2, E_2) . Finally, select the minimum-weight edge in E that crosses the cut (V_1, V_2) , and merge both minimum spanning trees into a single spanning tree.
Either prove that this algorithm is correct, or give an example for which this algorithm fails.
3. (16%) (a) Describe Floyd-Warshall algorithm for solving the all-pairs shortest path problems. (b) Prove that the algorithm is correct. (c) Briefly describe how to detect the presence of a negative-weight cycle based on the algorithm. (d) Reduce the transitive-closure problem to the problem of all-pairs shortest path.
4. (14%) Assume that we already know that the 3-CNT-SAT problem is NP-complete. Prove that the clique problem is NP-complete.

科目：演算法 B

日期：102 年 1 月 31 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.

For problem 3, and 4, you do not need to write the algorithm in a formal way. You may just describe the idea, use precise key word, and show the time complexity of your algorithm.

1. 5%

Give tight asymptotic upper bounds for $T(n)$ in each of the following recurrences.

Just give the answer, no need to give any explanation.

$$(1) T(n) = 2 T(n/2) + \log n$$

$$(2) T(n) = 2 T(n/2) + n \log n$$

$$(3) T(n) = 2 T(n/2) + n \log^2 n$$

$$(4) T(n) = 2 T(n/2) + n / \log n$$

$$(5) T(n) = 2 T(n/2) + n / \log^2 n$$

2.

(a) 2% Give a tight asymptotic upper bound for the following recurrence

$$T(n) = aT\left(\frac{n}{b}\right) + f(n) \quad \text{where } a > 1, b > 1, f(n) = \theta(n^{\log_b a})$$

(b) 8% Prove your answer.

3.

(a) 2% There is a theoretical lower bound on the worst case number of comparisons for any comparison sort algorithm. Give this lower bound, no need to give any explanation.

(b) 12% Show how to sort n integers in the range 0 to $n^2 - 1$ in $O(n)$ time.

(Describe the idea and explain the time complexity.)

(c) 3% Does the result in part (b) contradict to the fact in part (a)? Explain.

4. 18%

Give an efficient algorithm (polynomial time algorithm) to find the longest monotonically increasing subsequence of a sequence of n numbers.

◎請用深黑色鋼筆或原子筆出題

命題老師簽名：