九十九學年度第二次
博士班資格考

國立交通大學試題紙

科目：演算法(A)　　日期：100 年 7 月 28 日　

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. 16% Consider the comparison sorting problem.
   (A) The number of comparisons for any comparison sort algorithm has a theoretical lower bound in the worst case. Give that lower bound and show your answer.
   (B) Name three sorting algorithms whose time complexities achieve that lower bound. (Just give the name, no need to explain)

2. 17% The matrix-chain multiplication problem is to parenthesize a sequence of matrices so as to minimize the number of scalar multiplications.
   Now consider a variant of the matrix-chain multiplication problem in which the goal is to parenthesize the sequence of matrices so as to maximize, rather than minimize, the number of scalar multiplications.
   (A) Does this problem exhibit optimal substructure?
   (B) Design a polynomial time algorithm to solve this maximization problem.(Give the idea, formula, time complexity, etc.)

3. Assume the basic knowledge of NP-completeness: satisfiability , hamiltonian- cycle, vertex-cover, and subset-sum problems are NP-complete. And assume $P \neq NP$

   (1)4% Determine whether each of the following two problems (A) and (B) is in P or is NP-hard

   (2) 13% If it is in P, describe the algorithm briefly, and state the time complexity. If it is NP-hard, prove your answer.

   There are n programs and one storage device (say a tape or a disk). Let a_i be the length of the storage space needed to store the i th program, i= 1, 2, 3…,n. Let L be the length of the storage capacity of the storage device.

   (A) Determine the maximum number of these n programs that can be stored on one storage device (say a tape)with length L.

   (B) Determine the maximum total length of these n programs that can be stored on one storage device.

Note: The rule here to store a program on a tape is the following: A program has to be stored consecutively as a whole. One program cannot be split into several small parts and stored in different places, nor can be stored only part of it.

國立交通大學試題紙

九十九學年度第二次
博士班資格考

科目：演算法(B)　　日期：100年7月28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. For the following sub-questions, answer True / False

   Assume that function *f(n)* and *g(n)* are asymptotically non-negative. *T(n)* is the time an algorithm takes for an input of size *n*.

   A. (3%) $n^c = O(a^n)$ for constants c>0 and a>1.
   B. (3%) $\lg(n^c) = O(\lg n)$ for constant c>0
   C. (3%) For the two functions $f(n)$ and $g(n)$, either $f(n)=O(g(n))$ or $f(n)=\Omega(g(n))$
   D. (3%) The best case running time of Quicksort is $\theta(n)$
   E. (3%) For any uniform cost RAM program $T(n) = \Omega(S(n))$, where *S(n)* is the space an algorithm uses for an input of size *n*.

2. (5%) What is the reason that there is value in having so many different kinds of search tress (e.g., 2-3 trees, AVL trees, red-black trees, splay trees, …)?

3. (5%) Draw the binary search tree that results from inserting keys 10, 7, 8, 2, 42 in that order.

4. In Figure 1, the C++ structure Node represents a node in a linked-list. Please answer the following questions:

   A. (3%) If we pass the head node of a linked list as parameter p to *xyz*, what is the effect on the linked-list?
   B. (3%) Give the space complexity of running *xyz* in □ notation.
   C. (4%) Give a piece of C/C++ code that achieves the same effect as *xyz* with □(1) space complexity.

```
struct Node
{
    int v;
    Node *next;
};
```

```
void xyz(Node*& p) {
    if (!p) return;
    Node* r = p->next;
    if (!r) return;
    xyz(r);
    p->next->next = p;
    p->next = NULL;
    p = r;
}
```

Figure 1. xyz

5. (5%) Figure 2 shows the BELLMAN-FORD single-source shortest path algorithm. As a dynamic programming algorithm, BELLMAN-FORD maintains an optimal substructure throughout the execution of the main loop (Line 2~4). At the end of the i-th iteration of the main loop (Line 2~4), what is the corresponding optimal substructure?

   (**NOTE**: The optimal substructure is characterized by i and a few other parameters. Your answer has to clearly indicate the characterization with all relevant parameters. An ambiguous answer such as '*a shortest path has inherent optimal substructure*' will not get any point!)

```
BELLMAN-FORD (G, w, s)
1.  INITIALIZE-SINGLE-SOURCE (G, s)
2.  for i ← 1 to |V[G]| - 1
3.      do for each edge (u, v) ∈ E[G]
4.          do RELAX(u, v, w)
5.  for each edge (u, v) ∈ E[G]
6.      do if d[v] > d[u] + w(u, v)
7.          then return FALSE
8.  return TRUE
```

Figure 2. Bellman-Ford Shortest Path Algorithm

6. (10%) We want to solve the problem of determining the **longest-path distance** between two vertices in a **weighted directed acyclic graph**. A longest path is a simple path of maximum length in a given graph.

   A. Please describe a polynomial time algorithm that can determine the longest-path distance between two vertices in a weighted directed acyclic graph. Alternatively, if no such algorithm can be found, you need to prove that the problem is NP-Complete (or NP-Hard).
   B. If a polynomial time algorithm to the problem exists, please derive the worst-case time complexity of your algorithm in big O notation. You need to show the whole process of your derivation, not just the big O.