國立臺灣大學95學年度碩士班招生考試試題

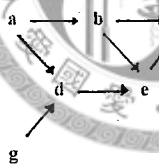
※ 注意：請於試卷上「非選擇題作答區」依序作答，並應註明作答之大題及小題題號。

1. (25 points)
   (a). Write a pseudo-code algorithm that builds a binary search tree from an array of positive integers. Imagine the array as a complete binary tree and transform the array into a binary search tree.
   (b). What is the time complexity of your algorithm?

   The functional prototype is as follows:

   ```
   int binTree[MAX_SIZE];
   buildBinTree(inout binTree[])
   // rearrange the elements of the array and make it as a binary search tree
   ```

2. (25 points)
   Consider a university that has K departments. Each department has its own undergraduate program in which a number of courses and their precedence order are defined. An example of course precedence order is shown as follows. Consider course and program as ADTs.



   (a) Write C++ class **definition** of ADT course. ADT course supports the following member functions: constructor, destructor, enQueue, and deQueue. enQueue enrolls a student into a course. deQueue takes a student from the course.
   (b) Write C++ class definition of ADT deptUndergraduateProgram that describes the courses and their precedence order under the program using pointer-based linked list. ADT deptUndergraduateProgram supports the following member functions: constructor, destructor, enQueue and deQueue. enQueue puts a new course into the program and specifies the proper position of the course in the precedence order graph. deQueue removes a course from the program.
   (c) Write C++ implementation of the "enQueue" of class deptUndergraduateProgram.

接背面

國立臺灣大學95學年度碩士班招生考試試題

3. (20 points)
An AVL tree is a binary search tree such that, for every node, the difference between the heights of its left and right subtrees is *at most* 1; the height of an empty tree is defined to be 0. Let $T(h)$ denote the number of nodes in a smallest AVL tree of height h (smallest in the sense of having the least number of nodes).

   (a) Define a recurrence relation for $T(h)$ ($h \geq 0$). Be sure to cover the base cases (or marginal cases).

   (b) Derive, based on the preceding recurrence relation, a lower bound for $T(h)$, showing that $T(h)$ grows at least exponentially with h. How do you infer, from the lower bound, the time complexity of performing a search operation on an AVL tree of size n?

4. (20 points)
Design an efficient algorithm that, given an array A of n integers and an integer x, determine whether A contains two integers whose sum is exactly x. Please present your algorithm in an adequate pseudo code and make assumptions wherever necessary. Give an analysis of its time complexity. The more efficient your algorithm is, the more points you will be credited for this problem.

5. (10 points)

   (a) What are NP problems? What are NP-hard problems? What are NP-complete problems?

   (b) Please describe as precisely as possible two example problems that are NP-complete. Why are NP-complete problems interesting (or worth studying)?

試題隨卷繳回