九十八學年度第一次
博士班資格考

國立交通大學試題紙

科目：演算法(A)　　日期：99 年 1 月 28 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%) We are given n points in the unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1, 2, \ldots, n$. Suppose that the points are uniformly distributed; that is, the probability of finding a point in any region of the circle is proportional to the area of that region. Design a $\Theta(n)$ expected-time algorithm to sort the n points by their distances $d_i = \text{sqrt}(x_i^2 + y_i^2)$ from the origin. (Hint: Design the bucket sizes in BUCKET-SORT to reflect the uniform distribution of the points in the unit circle.)

(Dynamic Programming)

2. (18%) (a) Describe an $O(n^3)$-time dynamic programming algorithm for finding an optimal parenthesization of a matrix-chain product. (b) Illustrate the algorithm by using an example of matrix-chain product whose sequence of dimensions is <5, 10, 3, 12, 5, 50, 6>. (c) 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. Does this problem exhibit optimal substructure?

(Greedy Algorithm)

3. (8%) Suppose a data file contains a sequence of 8-bit characters such that all 256 characters are about as common: the maximum character frequency is less than twice the minimum character frequency. Prove that Huffman coding in this case is no more efficient than using an ordinary 8-bit fixed-length code. (Hint: the maximum frequency of any two is less than twice that of any other two, and so on.)

(NP-completeness)

4. (14%) Prove that the 3-CNT-SAT problem is a NP-complete problem. (Hint: assume that SAT is already proved to be NP-complete.)

國立交通大學試題紙

九十八學年度第一次
博士班資格考

科目：演算法(B)　　日期：99 年 1 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. (10%) Binary search.
Let $A(1) < A(2) < A(3) < ... < A(n)$ be an array of n real numbers.
Given a number x, suppose that we use binary search to find whether x is in this array, and let T(n) be the worst case number of comparisons to determine whether in this array.
(1) Write a recurrence equation for solving T(n).
(2) Prove that the solution to this recurrence is $T(n) = \lfloor \log_2 n \rfloor + 1$

2. (10%) Give tight asymptotic upper bound for T(n). Just give the answer, no need to explain. Assume that T(n) is constant for sufficiently small n.
(1) $T(n) = 2T(\frac{n}{2}) + \frac{n}{\log_2 n}$
(2) $T(n) = 2T(\frac{n}{2}) + \frac{n}{(\log_2 n)^2}$
(3) $T(n) = 2T(\frac{n}{2}) + n(\log_2 n)^2$
(4) $T(n) \le 2T(4\sqrt{n}) + c \log n$
(5) $T(n) \le T(\frac{1}{5}n) + T(\frac{3}{4}n) + cn$

3. (10%) Minimum spanning tree.
Let e be a maximum-weight edge on some cycle of a graph G= (V, E). Prove that there is a minimum spanning tree of G'= (V, E – {e}) that is also a minimum spanning tree of G. That is, there is a minimum spanning tree of G that does not include e.

4. (8%) Binary search trees:
(1)What is the smallest possible height among all binary search trees containing n keys. (Say, n real numbers.) Prove your answer.
(2)What is the largest possible height among all binary search trees containing n keys. Justify your answer.

5. (12%) Suppose that a weighted, directed graph G= (V, E) has a negative-weight cycle.
   (1)Is there an efficient algorithm to find a shortest simple path between two vertices in such a graph? If so, describe briefly one such algorithm. If not, explain.
   (2) Is there an efficient algorithm to find one negative-weight cycle? If so, describe briefly one such algorithm. If not, explain.
   (3) Is there an efficient algorithm to find a negative-weight cycle with the most negative total weights? If so, describe briefly one such algorithm. If not, explain.