國立交通大學試題紙

九十八學年度第二次
博士班資格考

科目：演算法(A)　　　　日期：99年7月29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. 15%

Rank the following functions by order of growth; that is, find an arrangement $g_1, g_2, \ldots, g_n$, of functions satisfying $g_1 = \Omega(g_2)$, $g_2 = \Omega(g_3)$, ..., $g_{n-1} = \Omega(g_n)$. Partition your list into equivalence classes such that $f(n)$ and $g(n)$ are in the same class if and only if $f(n) = \Theta(g(n))$. (Note: $\log n = \log_{10} n$, $\lg n = \lg_2 n$, $\ln n = \log_e n$, $\log_e 2 = 0.693$ and $\log_{10} 2 = 0.301$).

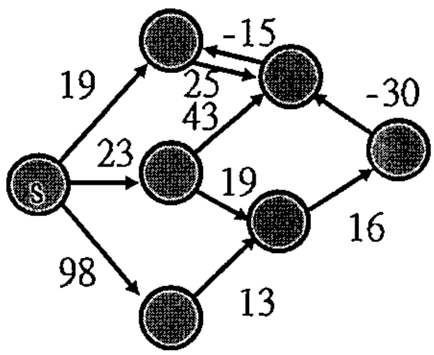
(a) $(\log n)^{2\log n}$
(b) $n^{\log \log^2 n}$
(c) $\log((n \log n)^{n \log n})$
(d) n^n
(e) $(1) + (1 + 1/2) + (1 + 1/2 + 1/3) + \ldots + (1 + 1/2 + \ldots + 1/n)$
(f) 2^{3n}
(g) 3^{2n}
(h) $n^2 + n^2/2 + n^2/4 + n^2/8 + \ldots + 1$
(i) $n + n/2^3 + n/3^3 + n/4^3 \ldots + n/(n^3)$
(j) $2\times(n-2)+4\times(n-4)+6\times(n-6)+\ldots+(n-2)\times 2$
(k) $1 + 1/2 + 1/3 + 1/4 + \ldots + 1/(100^n)$
(l) $2^{\lg n}$
(m) $4^{\lg n}$
(n) $8^{\lg n}$
(o) $8^{\log n}$

2. 10%

Given an open-address hash table with load faction $\alpha = n/m < 1$, prove that the expected number of probes in an unsuccessful search is at most $1/(1-\alpha)$, assuming uniform hashing.

3. 25%

a) Briefly describe Bellman-Ford Algorithm and illustrate it by the graph below. Note that you also need to describe that how to detect a graph with negative-weight cycle.
b) Briefly describe Dijkstra Algorithm and illustrate that it is not working in the same case.



國立交通大學試題紙

九十八學年度第二次
博士班資格考

科目：演算法(B)　　日期：99年7月29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. 8%
   (1) Write the recurrence equation to compute the average case time complexity of the quicksort and give the solution. (No need to explain.)
   (2) If you are asked to justify your solution, what method will you use? (No need to actually justify it, just state the procedure .)

2. 7% Suppose that you are given a sequence of n numbers to sort. The input sequence consists of n/k subsequences, each containing k numbers. The numbers in a given subsequence are all smaller than the numbers in the succeeding subsequence and large than the numbers in the preceding subsequence. Thus, all that is needed to sort the whole sequence of length n is to sort the numbers in each of the n/k subsequences. Prove that there is a theoretical lower bound on the number of comparisons needed to solve this variant of the sorting problem.

3. 10% Given that the subset-sum problem is NP-complete, prove that the set –partition problem is NP-complete.

The set –partition problem takes as input a set S of numbers. The question is whether the number can be partitioned into two set A and B= S – A such that the sum of the numbers in set A is equal to the sum of the numbers in set B. $\sum_{x\in A} x = \sum_{x\in B} x$

One of the following expressions may or may not be helpful:
$t-\sum_{x\in S} x$, $\left|t-\sum_{x\in S} x\right|$, $2t-\sum_{x\in S} x$, $\left|2t-\sum_{x\in S} x\right|$

4. 15% Show how to solve the subset-sum problem in polynomial time if the target value t is expressed in unary. (It is called pseudo-polynomial time. Name the method or technique used.)

5. 10%
   (1) The longest-simple-cycle problem is the problem of determining a simple cycle of maximum length in a graph. (The edge weight is one for each edge.)
   Is it an NP-hard problem, or is it a P problem? Justify your answer.
   (2) What about the shortest-simple-cycle problem?
   Is it an NP-hard problem, or is it a P problem? Justify your answer.