國立雲林科技大學資訊管理系

115 學年度第 1 學期博士班資格考

Department of Information Management
National Yunlin University of Science and Technology
Ph.D. Qualifying Examination, Semester 1, Academic Year 115

科目 Subject: **Algorithms**　　考試方式 Type: **Closed book**　　考試時間 Time: **3.5 Hours**

第一部份：(以下 6 題中請自行選擇 5 題作答；作答超過 5 題，僅計作答的前 5 題之分數)

Part 1: Please answer 5 of the following 6 questions.
If you answer more than 5 questions, only the first 5 answers will be graded.

1. Assume that an algorithm requires $f(n)$ microseconds to solve a problem of size n. One second contains 10^6 microseconds. Calculators are not allowed.
   (a) For each of the following running-time functions, determine the largest integer value of n for which the algorithm can finish within one second. **(6%)**
   $$\sqrt{n}, \quad n, \quad n^2, \quad n^3, \quad 2^n, \quad n!$$
   (b) Suppose that the computer becomes 1024 times faster. Explain how the largest solvable problem size changes for algorithms with running times n^2 and 2^n, respectively. **(4%)**

2. Consider the following sorted array using 1-based indexing:
   $A = [2,5,8,12,16,23,38,56,72,91,103,117,130,145,160]$.
   Use the binary search algorithm with $mid = \left\lfloor \frac{low+high}{2} \right\rfloor$.
   (a) Apply binary search to find the following target values. For each iteration, show low, high, mid, and A[mid]. State whether the target is found and, if found, give its position. **(6%)**
      1. Target 91
      2. Target 100
   (b) Write and solve the worst-case running-time recurrence of binary search. **(4%)**

3. Given the array A = [38,27,43,3,9,82,10,15], answer the following questions:
   (a) Use merge sort to sort the array in ascending order. Show the division and merging process, and give the final sorted array. **(7%)**
   (b) Write the worst-case running-time recurrence of merge sort and state its asymptotic running time. **(3%)**

4. Consider a hash table with 10 slots indexed from 0 to 9. The hash function is $h(k) = k \bmod 10$, and collisions are resolved using linear probing. Insert the keys 23, 43, 13, 27, 37, and 17 in the given order, and show the final hash table. **(10%)**

5. Given the max heap $A = [16,14,10,8,7,9,3,2,4,1]$, using 1-based indexing, perform MAX-

HEAP-INSERT(A, 15), followed by HEAP-EXTRACT-MAX. Show the array after each exchange, and give the extracted key and the final max heap. **(10%)**

6. Please explain the differences between (a) P (Polynomial) (b) NP (Nondeterministic Polynomial) (c) NPC (NP-complete), and (d) NPH (NP-hard) in terms of problems that can be decided, verified or reduced in polynomial time. **(3% each, max 10%)**

第二部份：(以下 2 題中請自行選擇 1 題作答；作答超過 1 題，僅計作答的第 1 題之分數)

Part 2: Please answer 1 of the following 2 questions.
If you answer more than 1 question, only the first answer will be graded.

7. Given a binary search tree with N nodes, design an algorithm to find the k-th largest element in the tree. Assume that the value of k is valid, i.e., $1 \leq k \leq N$. The algorithm must run in O(h) time, where h is the height of the tree. Provide a detailed step-by-step explanation of the algorithm using pseudocode. Prove the correctness of the algorithm and analyze its time complexity. **(14%)**

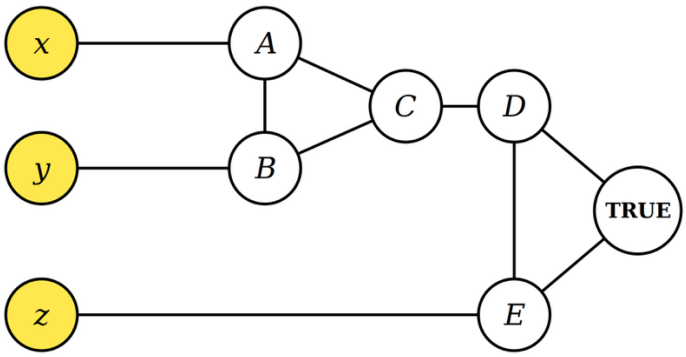
8. To reduce the 3-CNF-SAT problem to a 3-COLOR (TRUE, FALSE, or RED) problem, the clause $(x \vee y \vee z)$ is represented as the following graph. Demonstrate that if each of x, y and z is colored either c(TRUE) or c(FALSE), then the widget is 3-colorable if and only if at least one of x, y, or z is colored c(TRUE). **(14%)**

(Hint: $c(u) \neq c(v)$ for every edge $(u, v) \in E$)

The edge set of the graph is:

x–A, y–B, A–B, A–C, B–C, C–D, D–E, D–TRUE, E–TRUE, z–E

The vertex labelled TRUE has its color fixed at c(TRUE).



第三部份：(以下 4 題中請自行選擇 3 題作答；作答超過 3 題，僅計作答的前 3 題之分數)

Part 3: Please answer 3 of the following 4 questions.
If you answer more than 3 questions, only the first 3 answers will be graded.

9. (Amortized analysis) Consider a k-bit binary counter that starts at 0. The cost of an operation is the number of bit flips it performs.

(a) Using the potential method, define a suitable potential function Φ and prove that the amortized cost of INCREMENT is O(1). **(5%)**

(b) Suppose the counter must also support DECREMENT. Construct a sequence of n operations whose total actual cost is $\Theta(nk)$. **(4%)**

(c) Continuing from (b), prove that for a counter supporting both INCREMENT and DECREMENT there exists no valid potential function that establishes an O(1) amortized cost. **(3%)**

10. (Theoretical properties of minimum spanning trees)

Notation for this question. A cut $(S, V-S)$ of an undirected graph $G = (V, E)$ is a partition of V. An edge (u, v) crosses the cut if one of its endpoints lies in S and the other in $V-S$. A cut respects a set A of edges if no edge of A crosses it. An edge is a light edge crossing a cut if its weight is the minimum among all edges crossing that cut (there may be more than one in case of ties). An edge (u, v) is safe for A if $A \cup \{(u, v)\}$ is still a subset of the edges of some minimum spanning tree.

(a) Let A be a subset of the edges of some minimum spanning tree, let $(S, V-S)$ be a cut that respects A, and let (u, v) be a light edge crossing this cut. Prove that (u, v) is a safe edge for A. **(5%)**

(b) Prove that if all edge weights in a graph are distinct, then the minimum spanning tree is unique. **(4%)**

(c) Define the bottleneck value b(T) of a spanning tree T to be the largest edge weight in T. Prove that every minimum spanning tree is a minimum bottleneck spanning tree, and give a counterexample showing that the converse does not hold. **(3%)**

11. (Maximum flow and bipartite matching) Let $G = (L \cup R, E)$ be a bipartite graph.

(a) Explain how to reduce the maximum bipartite matching problem on G to a maximum-flow problem on a flow network, and prove that the value of a maximum flow in that network equals the size of a maximum matching. Your answer must explain the role played by the integrality theorem. **(6%)**

(b) Let M be a matching in a graph $G = (V, E)$.

Notation for this question. For sets A and B, the symmetric difference is $A \oplus B = (A - B) \cup$

(B − A), the elements belonging to exactly one of them. A vertex incident on an edge of M is matched under M, and otherwise unmatched. An M-alternating path is a simple path whose edges alternate between M and E − M; an M-augmenting path is an M-alternating path whose first and last edges both belong to E − M.

(i) Prove that if P is an M-augmenting path, then M ⊕ P is also a matching in G and |M ⊕ P| = |M| + 1. **(2%)**

(ii) Let M and M′ be matchings in G with |M′| > |M|. Prove that every connected component of M ⊕ M′ is an isolated vertex, a simple path whose edges alternate between M and M′, or an even-length simple cycle whose edges alternate; and deduce that it contains at least |M′| − |M| vertex-disjoint M-augmenting paths. **(2%)**

(iii) Using (i) and (ii), prove that M is a maximum matching if and only if G contains no M-augmenting path. **(2%)**

12. (Approximation algorithms and inapproximability)

(a) Prove that APPROX-VERTEX-COVER is a 2-approximation algorithm for the vertex-cover problem. **(4%)**

```
APPROX-VERTEX-COVER(G)
1   C = ∅
2   E′ = G.E
3   while E′ ≠ ∅
4       let (u, v) be an arbitrary edge of E′
5       C = C ∪ {u, v}
6       remove from E′ every edge incident on either u or v
7   return C
```

(b) Construct a family of instances on which the algorithm returns a solution exactly twice the size of the optimal one, thereby showing that the approximation ratio of 2 is tight. (**(3%)**

(c) Prove that if P ≠ NP, then for any constant ρ ≥ 1 there is no polynomial-time ρ-approximation algorithm for the general travelling-salesperson problem. Briefly explain why this conclusion fails when the weights satisfy the triangle inequality. **(5%)**

— End of examination —