國立中山大學資訊工程學系
104 學年度第 1 學期博士班資格考試

科目：演算法

1. Suppose we obtain the following recurrence formula of time complexity for solving a problem:
   $T(n)=b$, if $n \leq 2$
   $T(n)=2T(n/2)+cn$, if $n > 2$,
   where n is the input size of the problem, b and c are constants. Please derive the time complexity and represent it with O notation. (15%)

2. Present an algorithm for solving the 2-D maxima finding problem with O(n log n) time. You should analyze your time complexity. (Hints: The divide-and-conquer strategy is a good approach.) (15%)

3. In the solution tree searching strategy, explain each of the following and give the data structure used by each. (15%)

   (a) depth-first search

   (b) breadth-first search

   (c) best-first search

4. (a) The selection problem is to select the kth smallest element among n input elements. Please design an algorithm to solve the problem. The time required for your algorithm should be O(n). (Hints: The prune-and-search strategy is a good approach.) (10%)

   (b) What is the general recurrence form for computing the time complexity of a prune-and-search algorithm (not particular for the selection problem)? (5%)

5. (a) Explain the longest common subsequence (LCS) problem. And, then give an example to illustrate your answer. Note that you should give both explanation and example. (5%)

   (b) Give the dynamic programming method for calculating the LCS length. (5%)

6. Prove that the *clique* decision problem polynomially reduces to the *node cover* decision problem. (15%)

7. Design an algorithm to determine whether a graph $G=(V,E)$ is 2-colorable. Your algorithm should be done in O($n+m$) time, where $n=|V|$ and $m=|E|$. Give an example to illustrate your algorithm and explain why your algorithm can be performed in O($n+m$) time. (15%)