Dept. of Computer Science and Engineering, National Sun Yat-sen Univ.
Second Semester of 2013 PhD Qualifying Exam Computer Algorithms

1. Suppose we obtain the following recurrence formula of time complexity for solving some problem:

$$T(n) = \begin{cases} b & \text{if } n \leq 2 \\ 2T(\frac{n}{2}) + cn & \text{if } n > 2, \end{cases}$$

where n is the input size of the solved problem. Please derive the time complexity and represent it with O notation. (15%)

2. Present an algorithm for solving the *minimum spanning tree* problem of a graph. And analyze the time complexity of your algorithm. (15%)

3. (a) What is the difference between *divide-and-conquer* and *prune-and-search*? (5%)

   (b) Is the *binary search* algorithm a divide-and-conquer method or a prune-and-search method? Why? (5%)

4. 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*

5. In the self-organizing sequential search heuristics, what are the *transpose heuristics, move-to-front heuristics* and *count heuristics*? (15%)

6. Show that the *Hamiltonian cycle* problem reduces to the *traveling salesperson decision* problem. (15%)

7. Given a set S of n positive integers, the 4-subset problem is to determine whether there exist four distinct elements a, b, c and d in S such that $a + b + c = d$. Please design an algorithm to solve this problem, and analyze the time complexity of your algorithm. (15%)

   (a) If the time complexity of your algorithm is $O(n^3 \log n)$, then you will get only 5% points.

   (b) If the time complexity of your algorithm is $O(n^2 \log n)$, then you will get the full 15% points.