Dept. of Computer Science and Engineering, National Sun Yat-sen Univ.
Second Semester of 2012 PhD Qualifying Exam Computer Algorithms

1. Explain each of the following terms. (16%)

   (a) *NP, NP-complete*

   (b) *selection problem*

   (c) *amortized time analysis*

   (d) *randomized algorithm*

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

$$T(n) = \begin{cases} b & \text{if } n \le 2 \\ 4T(\frac{n}{2}) + cn & \text{if } n > 2, \end{cases}$$

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

3. Present an algorithm for solving the *2-D maxima finding* problem with $O(n \log n)$ time. You should analyze your time complexity. (15%)

4. (a) In the searching strategy, explain *breadth-first search*, *depth-first search*, *best-first search* and *hill climbing.* (8%)

   (b) What data structures are used the breadth-first search and depth-first search, respectively? (4%)

5. (a) What is an *AVL* tree? (3%)

   (b) In what situation will an AVL tree increase its height? (3%)

   (c) Is a new node in an AVL tree always inserted as a leaf node? Why? (3%)

   (d) When a node in an AVL tree is to be deleted, is it always a leaf node? Why? (3%)

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

7. For given any sequence of four numbers, we can apply the simple *bubble sort* to them to get a sorted sequence in six comparison operations. Please design a special sorting algorithm for four numbers. Your algorithm should sort any sequence of four numbers in five comparison operations. Here, data movements are not counted. (15%)