Dept. of Computer Science and Engineering, National Sun Yat-sen Univ.
First Semester of 2005 PhD Qualifying Exam Computer Algorithms

1. Explain the following terms. (20%)
   (a) *greedy method*
   (b) *1-center problem*
   (c) *bi-connected graph*
   (d) *randomized algorithm*
   (e) *exact set cover problem*

2. (a) Explain the *longest common subsequence* (LCS) problem. Give an example for illustration. (5%)
   (b) Give the *dynamic programming* approach to solve the LCS problem. What is the time complexity of this algorithm? Why? (10%)

3. (a) Use the *prune-and-search* approach to design an algorithm for selecting the kth smallest element among n input elements. Your algorithm should be with $O(n)$ time. (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%)

4. The *bin packing decision* problem is defined as follows. We are given a set of n items, each of size c_i, which is a positive integer. The capacity of each bin is a positive integer M. The problem is to determine whether we can assign items into k bins, where k is a constant, such that the sum of $c_i's$ assigned to each bin does not exceed M. Prove that the *partition* problem reduces to the bin packing decision problem. (15%)

5. In doing the sorting job, suppose only one operation $M_{i,j}$ can be used. For given a permutation $P = a_1, a_2, \cdots, a_n$, if $M_{i,j}$, $1 \leq i \leq j \leq n$, is performed, the substring between positions i and j is moved to the front. In other words, $M_{i,j}(P) = P' = a_i, a_{i+1}, \cdots, a_j, a_1, a_2, \cdots, a_{i-2}, a_{i-1}, a_{j+1}, a_{j+2}, \cdots, a_n$.
   (a) Suppose the input permutation is 3476125. How do you sort it into 1234567 by only using a sequence of M operations? You have to give the M operations you use. (10%)
   (b) What is the worst input permutation of 7 distinct numbers, which requires the maximum number of M operations to sort the input into increasing order? Why? (10%)

6. There is a C function in the following:

```
int a[ ]={23,26,21,25,27,28,22,20};
int f(int low, int high)
{
   int mid, x, y, m;

   if (high-low ==0)
      m=a[high];
   else
   {
      mid = (int) (high+low)/2;    /* e.g. 7/2=3, 6/2=3 */
```

```
        if ((x=f(low, mid) > (y=f(mid+1, high)))
            m=x;
        else
            m=y;
    }
    printf("%d   %d \n",high-low+1, m);
    return m;
}
void main( )
{
    int n=8;

    f(0, n-1);
}
```

(a) What will be printed after the program is executed? (5%)

(b) Suppose in the *main*( ) function, the value of n is given arbitrarily, where $n = 2^k$, k is a positive integer constant, and the array $a[\ ]$ also has n elements. How many lines will be printed after the program is executed? You should derive the formula. (5%)

(c) If the program segment

```
        if ((x=f(low, mid) > (y=f(mid+1, high)))
            m=x;
        else
            m=y;
```

in the above program is replaced by the following segment

```
        if (f(low, mid) > f(mid+1, high))
            m=f(low, mid);
        else
            m=f(mid+1, high);
```

answer question (b) again. (5%)