# I500/B609: Fundamental Computer Concepts of Informatics

HW1 (Due: **Sep. 9 Friday BEFORE** Lab session)

http://darwin.informatics.indiana.edu/col/courses/I500-16

1. (10 pts) Illustrate the operation of Insertion-sort algorithm on array A=<31,41,59,26,41,58>.

2. (10 pts) The input to the algorithm *Unknown* illustrated below is an array A of N numbers. (1) what is the output of the algorithm? (2) using big-O notation to show the running time of the algorithm.

```
Input: Array A of N numbers;
Unknown(A)
for j = 1 to N-1
        if A[N] < A[j]
                exchange A[j] and A[N]
Output A[N];
```

3. (Analysis of selection-sort algorithm, 20 pts)

Selection-sort is an algorithm to sort a given list of numbers. It works as follows:

1. Find the minimum value in the list;
2. Swap it with the value in the first position;
3. Repeat the steps above for the remainder of the list (starting at the second position and advancing each time)

```
SELECTION-SORT(A)
n = A:length
for j = 1 to n-1
        smallest = j
        for i = j+1 to n
                if A[i] < A[smallest]
                        smallest = i
        exchange A[j] with A[smallest]
```

What is the worst case and best case running time of selection-sort? How does it compare to insertion sort?

4. (20 pts) An array A[1..n-1] contains all integers from 0 to n except two numbers. It would be easy to determine to the two missing numbers by using an auxiliary array B[0..n] to record which numbers appear in A. Here, we want to avoid the additional storage of B with the size O(n). Devise an algorithm to determine the two missing integers in O(n) time under this constraint. (Note, you can still use additional constant memory as temporary storage.)

5. (10 pts) Let A[1..n] be an array of *n* distinct numbers. If i < j and A[i] > A[j], then the pair (i, j) is called an inversion of A. a) List the five inversions of the array <2,

3, 8, 6, 1>. b) What array with elements from the set {1, 2, …, n} has the most inversions? How many does it have? c) What is the relationship between the running time of insertion sort and the number inversions in the input array? Justify your answer.

6. (20 pts) You are given two sorted lists of size m and n. Devise an O(log m + log n) time algorithm for computing the *k*th smallest element in the union of the two list.

7. (10 pts) Given an array of numbers as the input, devise an algorithm to generate a random permutation of the array, such that each number has equal probability to be placed in each position in the output array.