College of Computer & Information Science — Fall 2006
Northeastern University — Handout 17
CSU690: Algorithms and Data — 10 November 2006

Problem Set 6 (due Tuesday, November 21)

1. (12 points) Network flows and atmospheric science experiments

Chapter 7, Exercise 20, page 426.

2. ($4 \times 4 = 16$ points) Average-case analysis of insertion sort

Insertion sort is a simple iterative sorting algorithm, the pseudocode for which is given below.

```
InsertionSort(A)
for j ← 2 to n
    key = A[j]
    i ← j − 1
    while i > 0 and A[i] > key
        A[i + 1] ← A[i]
        i ← i − 1
    A[i + 1] ← key
```

(a) Prove that the worst-case running time for insertion sort is $\Theta(n^2)$.

While the worst-case running time of both insertion sort and quicksort is $\Theta(n^2)$, quicksort is considered an efficient sorting algorithm because the randomized version of quicksort has an expected running time of $\Theta(n \lg n)$ (as we have seen in class). We now consider how insertion sort performs on average.

Let $A[1..n]$ be an array of distinct numbers. If $i < j$ and $A[i] > A[j]$, then we say that the pair (i, j) is an *inversion* of A.

(b) When does an array have the minimum number of inversions? What is this minimum number? When does an array have the maximum number of inversions? What is this maximum number?

(c) If $T(A)$ is the running time of insertion sort on input A and $I(A)$ is the number of inversions in A, then prove that $T(A)$ is $\Theta(I(A) + n)$.

(d) Show that if A is a random permutation chosen uniformly at random from the set of all $n!$ permutations, then the expected number of inversions is $\Theta(n^2)$. Now invoking part (c), derive a tight bound on the expected running time of insertion sort on a random permutation.

3. (12 points) Random generation of peer-to-peer networks

Chapter 13, Exercise 4, page 784.

4. (12 points) One-pass auctions

Chapter 13, Exercise 9, page 788.

5. (8 points) Co-op experience

Describe the project that you were involved with during your most recent co-op assignment. Did you encounter an algorithmic problem during this assignment, directly or indirectly? If so, try to formulate it as a formal algorithms problem. (A negative answer is perfectly acceptable.)

Your answer must be typed and be 1-2 pages long, using reasonable margins and line-spacing. Please turn in the answer as a separate sheet.

Bonus Problem. (6 points) A fast network flow algorithm

Chapter 7, Exercise 11, page 420.