1. **Problem 8.3-2 (p. 173) (30 Points)** Which of the following algorithms are stable: insertion sort, merge sort, heapsort, and quicksort? Give a simple scheme that makes any sorting algorithm stable. How much additional time and space does your scheme entail?

2. **Problem 8.3-4 (p. 173) (25 Points)**
   Show how to sort \( n \) integers in the range 0 to \( n^2 - 1 \) in \( O(n) \) time. *Hint: Use two digit radix-n sort*

3. **Problem 9.3-3 (p. 192) (20 Points)**
   Show how quicksort can be made to run in \( O(n \log n) \) time in the worst case.

4. **Problem 9.3-5 (p. 192) (25 Points)**
   Suppose that you have a “black-box” worst-case linear-time median subroutine. Give a simple, linear-time algorithm that solves the selection problem for an arbitrary order statistic.

**Bonus Problem Due by Thursday October 17, 2002**

1. **Problem 9.3-1 (p. 192) (30 Points)**
   In the algorithm `SELECT`, the input elements are divided into groups of 5. Will the algorithm work in linear time if they are divided into groups of 7? Argue that `SELECT` does not run in linear time if groups of 3 are used.