Problem 1. Assume you have a sorted list of $n \geq 2$ distinct values in which exactly two different elements have been interchanged. Assume you execute insertion sort with a sentinel on this list. For each of the following justify your answer.

(a) Assume that the elements in positions i and j were interchanged, where $i < j$. Derive a general formula for the number of comparisons (as a function of i and j).

(b) Derive a general formula for the number of moves (as a function of i and j).

(c) What is the best case number of comparisons (over all pairs i and j)?

(d) What is the best case number of moves (over all pairs i and j)?

(e) What is the worst case number of comparisons (over all pairs i and j)?

(f) What is the worst case number of moves (over all pairs i and j)?

(g) What is the average case number of comparisons?

(h) What is the average case number of moves?

Problem 2. Assume you have a sorted list of $n \geq 3$ distinct values in which exactly two different elements have been interchanged. For each of the following justify your answer.

(a) Give an efficient algorithm to determine which two elements were interchanged. Give the pseudo code. Minimize the *exact* number of comparisons in the *worst* case.

(b) Assume that the elements in positions i and j were interchanged, where $i < j$. Derive a general formula for the number of comparisons as a function of i and/or j and/or n. (It should not be a function of only n.)

(c) How many comparisons does your algorithm use in the best case?

(d) How many comparisons does your algorithm use in the worst case?

(e) How many comparisons does your algorithm use in the average case?

(f) What is the (exact) high order term for the average case?

(g) **Challenge Problem.** Give an algorithm that minimizes the average number of comparisons, and analyze it.