The following algorithm for sorting is called slowsort:

\[
i ← 1 \\
\text{while } i < n \text{ do} \\
\quad \text{if } A[i] > A[i+1] \text{ then} \\
\quad \quad A[i] ↔ A[i+1] \\
\quad \quad i ← 1 \\
\quad \text{else} \\
\quad \quad i ← i + 1 \\
\text{end if} \\
\text{end while}
\]

Problem 1.

(a) What is the best-case input for comparisons for slowsort. (No justification needed.)
(b) What is the exact number of comparisons for slowsort in the best case. (No justification needed.)
(c) What is the high order term?
(d) Write the best-case running time in order notation.

Problem 2.

(a) What is the worst-case input for comparisons for slowsort. (No justification needed.)
(b) Write a summation (actually nested summations) for how many comparisons slowsort uses in the worst case. Hint 1: The algorithm has a final pass to confirm that the list is sorted. Leave this pass out of your summation (but do not drop the term). Hint 2: Try some small examples (n=1,2,3,4 etc.).
(c) Simplify your summation.
(d) What is the high order term?
(e) Write the worst-case running time in order notation.

Problem 3.

(a) Write a summation (actually nested summations) for how many comparisons slowsort uses in the average case.
(b) Simplify your summation.
(c) What is the high order term?
(d) Write the average-case running time in order notation.