1. Work in groups of 2–4 people. List the members of your group below. Underline your name.

2. *Double-ended insertion sort* is an algorithm that is essentially two insertion sorts operating in an interleaved manner, one at each end of the array being sorted. We may describe the algorithm as follows. Two agents, A and B, start at the low-index and high-index ends of the array, respectively. Each has an associated region of the array, called its *territory*. Elements in A's territory are in ascending order while those in B's territory are in descending order (reading left to right). Initially, A's territory is an empty region to the left of the first element of the array and, similarly, B's territory is an empty region to the right of the last element. The part of the array that is in neither A's nor B's territory is called *open territory*. The agents take turns and operate as follows until the open territory is empty:

   - Agent A: Let x be the leftmost element in the open territory. Use a sequential scan to find y: the rightmost element in A's territory that is less than or equal to x. (If no such y exists then let y be an imaginary element to the left of the leftmost real element.) Move each element in A's territory that is to the right of y one position to its right and place x in the now vacant position immediately to the right of y. As a result, A's territory has grown to the right by one element.
   - Agent B: Perform actions similar those of A above, but replace A by B, less by greater, and right by left.

   (a) Using the textbook's[1] notation and conventions, provide detailed[2] pseudocode for the above algorithm.

   (b) Provide a formal statement of correctness of the pseudocode.

   (c) Use loop invariants and related constructs to prove the above statement, using the textbook's proof of insertion sort [3] as a guide.

   (d) Use the textbook's method [4] to determine a detailed expression for the running time of the pseudocode as a function of the number of elements in the array.

   (Use next page and additional paper as needed.)

[1] Thomas H. Cormen et al., *Introduction to Algorithms*, 4th edition (MIT Press, 2022), Section 2.1, esp. pp. 21–24.

[2] It should be trivial for a first-year programmer to map the pseudocode to C, Java, or Python code.

[3] Cormen et al., *op. cit.*, Section 2.1, pp. 19-21.

[4] *Idem*, Section 2.2, esp. pp. 30–31.

3. Briefly summarize the contributions of each member of the group.

4. *Informal* homework:

   (a) Implement, test, document, and the algorithm. Package the implementation along with suitable tests.

   (b) Conduct and document a detailed experimental evaluation of the above implementation. Explain any variations from the performance predicted by the earlier analysis.

   (c) (554) How may the algorithm be modified (as little as possible) to enable agents A and B to operate in parallel efficiently? Redo all the other questions for this version of the algorithm.