

Name: _____

1. (1 pt.)

- **Read all material carefully.**
- You may refer to your books, papers, and notes during this exam.
- No computer or network access of any kind is allowed (or needed).
- Write, and draw, carefully. Ambiguous or cryptic answers receive zero credit.
- Use the conventions used in class and the textbook for notation, algorithmic options, etc.

Write your name in the space provided above.

2. (9 pts.) For each value of $n = 2, 3, 4$, depict all possible AA-trees containing the n keys $1, 2, \dots, n$. *Justify your answer* briefly, explaining why the trees you depict are the only ones possible.

3. (10 pts.) For each tree of Question 2: (1) provide a sequence insertions that produces the tree and (2) depict the result of each insertion, clearly indicating any rotations.

[additional space for answering the earlier question]

4. (5 pts.) What is the minimum number of nodes in an AA-tree with three levels? Justify your answer. Depict a 3-level AA-tree with the claimed number of nodes.

5. (5 pts.) Provide a sequence of insertions that produces the tree of Question 4. Depict the details of these insertions, as in Question 3.

[additional space for answering the earlier question]

6. (10 pts.) Repeat Question 4 replacing *minimum* with *maximum*. Provide a sequence of insertions that produces this tree and *explain clearly* why it works. (You do not need to depict the insertion details as in Question 5.)

[additional space for answering the earlier question]

7. (20 pts.) *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.

Provide Java code for a static method that uses the above algorithm to sort (in place) a given array of `ints`. Follow the algorithm as closely as possible and do not use Java library methods. Ensure that the correctness of your code is evident with the aid of suitable comments; cryptic code will receive no credit.

[additional space for answering the earlier question]

8. (10 pts.) For each tree of Question 2, list *all* sequences of insertions that produce that tree. Credit requires that you explain clearly why each sequence produces the tree you claim, but you do not need to depict the details of the insertions.