Written Portion

W1 **(10 Points) Weiss 2.1**

- -1: If one element is out of order, or two elements are swapped
- -5: if four of the elements are out of order
- -6: if six of the elements are out of order
- -8: if MORE THAN six of the elements are out of order
- Additionally, -2 if wrong answer for which two have the same rate
- Additionally, -2 if 1 or more elements are missing in the answer

W2 **(10 Points) Weiss 2.6**

NOTE: Soln is actually 2^(2^(x-1)) for Part A, adjust Part B accordingly

- + 5 points for part A
  - + 4 points if they get **2^(2^(x))**, and they don't specify that first day = 0
  - + 2 points are awarded if wrong answer but explanation is given
- + 5 points for part B. Accepting both log log N and 1 + log log N.
  - + 4 points if answer follows from A, even if A is incorrect (unless A was 2^(2^(x)))
  - + 2 points are awarded if wrong answer but explanation is given
  - 0 points are awarded if wrong answer and no work shown.

W3 (10 Points)

- + 3 points for each answer if it is correct.
  - + 1 point given if wrong answer but work shown.
  - 0 points given if wrong answer and no points.
  - The additional 1 point is automatically given if all three parts of the question are attempted

  Note: For a), accept both O(23n) and O(n).

W4 (10 Points) Weiss 2.11

- -2 if one wrong; -5 if two wrong; -7 if three wrong
- -1: if minor error, but work shown with correct method, e.g: rounding off error (such as giving 12.5 as 12), or computation error (*not* mathematical manipulation errors)

W5 (10 Points) Weiss 2.15

Algorithm

- +6 points for algorithm that is optimal.
- +5 points for algorithm that is optimal but contains implementation errors in description
- +3 points if correct algorithm given but not the best possible (linear vs binary).
- +2 points if problem is wrong but attempt shown.
- 0 points if no work

Runtime

- +4 points for the running time of the algorithm.
- +2 points if work shown on why they got the running time but it is incorrect.
- 0 points for wrong answer and no work.

Programming Portion

- ½ if the code doesn't compile and major changes are needed. (Major changes: not immediately identifiable logic errors, **multiple** syntax errors)
- (-2) if program doesn't compile due to minor syntax errors (e.g missing semicolons)
- (-1) for not removing package statements.

P1(18 Points):

Rectangle class (10):
+ 2 points for writing out getLength and getWidth
+ 2 points for correct implementation

+ 3 points Implemented a compareTo method ()
+ 3 points for correct implementation.

Problem1 class (8):
+ 4 points for correctly creating an array of rectangle objects.
    - 2 for array of rectangles
    - 2 for populating it properly.
        (-1 if student fails to populate the array with 5 rectangle objects or more)
        (1 only if trivial population)
+ 4 points for correctly calling findMax and returning max.

P2 (18 Points)

Main Method (4pts):
(+1 points) Successfully create array of rectangle objects
(+1 point) Correctly calling binarySearch method
(+2 points) Arrays.sort() called

Binary Search (14pts):
(+3 points) Using correct compare method
(+3 points) Using generics
(+4 points) Correctly implementing binarySearch using recursion

- (-1) for each index incorrectly set during a recursive call
- (-2) if binarySearch doesn't return -1 for Rectangle is not found

(+2 points) Using correct method signature
(+2 points) Using efficient algorithm for binarySearch

NOTE:
(-1 point) If use while (start < end) instead of if (To make them aware of difference between iteration and conditional)

(-2 points) If they use global variables to track start and end values instead of creating a helper method (no way to reset start and end after binarySearch is called once)

P3(14 Points)

Copying Code (6 points):
(+2 points) Copies method 1 correctly from written part 3
(+2 points) Copies method 2 correctly from written part 3
(+2 points) Copies method 3 correctly from written prat 3

Measuring Time (4 points)
(+2 points) Uses System.currentTimeMillis() or System.nanoTime correctly, sandwiching method to be measured
(+2 points) Records results in some manner (using File IO or System.out)

Explanation (4 points)
(+4 points) Describes growth patterns and links them to Big-Oh

- (1 Point) First fragment = O(N)
- (1 Point) Second fragment = O(N^2)
- (2 Points) Third fragment = O(log N)