1.) Main work is in the program NodeSequence.java.

2.) a. Compare two elements, and set min to the smaller number and max to the larger number. Continue comparing elements, with then comparing the smaller of the two to the min, and the larger of the two to the max. Because you do not have update min and max in the first step, the overall comparisons will be (3n/2) - 2.

b.) Look at each row of the array, effectively each as a one-dimensional array. Cut the array in half. If the left array's last element is a 1, add the length of the array to a variable, and then look at the right array. If the right array starts with a 0, you have the number of 1s in the array row. If the left array ended in a zero, you repeat the cutting process, only adding the length of the left arrays if they end in zero. The same process is done with right arrays that start with 1. Once you find the place where the left array ends in 1 and the right array ends in 0, you are finished. Repeat for all rows, and then add together values to obtain the number of 1s.