ALGORITHM
Due Date: 9:20AM, October 18
Autumn, 2012
The following problem sets are all from CLRS.

Homework 2

1. Using the figure in page 6 of Lecture Note 2 as a model, illustrate the operation of INSERTION-SORT on the array $A = \langle 31, 41, 59, 26, 41, 58 \rangle$.

2. Using the figure in page 17 of Lecture Note 2 as a model, illustrate the operation of MERGE-SORT on the array $A = \langle 3, 41, 52, 26, 38, 57, 9, 49 \rangle$.

3. Bubble sort is a popular sorting algorithm. It works by repeatedly swapping adjacent elements that are out of order.

```
BUBBLE-SORT($A$)
1  for $j \leftarrow 1$ to $length[A]$
2      do for $j \leftarrow length[A]$ downto $i + 1$
3             do if $A[j] < A[j-1]$
4                   then exchange $A[j] \leftrightarrow A[j-1]$
```

(a) Let A' denote the output of BUBBLE-SORT(A). To prove that BUBBLE-SORT is correct, we need to prove that it terminates and that

$$A'[1] \leq A'[2] \leq \ldots \leq A'[n], \tag{1}$$

where $n = length[A]$. What else must be proved to show that BUBBLE-SORT actually sorts?

The next two parts will prove inequality (1).

(b) State precisely a loop invariant for the **for** loop in lines 2–4, and prove that this loop invariant holds. Your proof should use the structure of the loop invariant proof presented in this chapter.

(c) Using the termination condition of the loop invariant proved in part (b), state a loop invariant for the for loop in lines 1–4 that will allow you to prove inequality (1). Your proof should use the structure of the loop invariant proof presented in this chapter.

(d) What is the worst-case running time of BUBBLE-SORT? How does it compare to the running time of INSERTION-SORT?