1. Recall Stooge-Sort from homework assignment 3:

\[
\text{Stooge-Sort (A,i,j)} \\
1 \text{ if } A[i] > A[j] \\
2 \quad \text{swap } A[i] \text{ and } A[j] \\
3 \text{ if } i+1 >= j \\
4 \text{return} \\
5 \quad k = \text{floor}((j-i+1)/3) \quad \text{// round down} \\
6 \text{Stooge-Sort (A,i,j-k)} \quad \text{// First two-thirds} \\
7 \text{Stooge-Sort (A,i+k,j)} \quad \text{// Last two-thirds} \\
8 \text{Stooge-Sort (A,i,j-k)} \quad \text{// First two-thirds again}
\]

Complete the following proof of the correctness of Stooge-Sort.

Proof is by induction on the size of the array A.

(base case: 1 and 2 element arrays) When the length of input is 1, Stooge-Sort does nothing before returning, which is correct since an array of size 1 is already sorted. When the length of input is 2, it puts these 2 elements into the correct order, then returns.

Induction Hypothesis: Stooge-Sort sorts correctly for input of any size between 1 and n - 1.

(induction step) Consider input of size n: A = (a_1, a_2, ..., a_n), where n ≥ 3.
By the induction hypothesis, lines 6, 7, and 8 each correctly sort two-thirds of A. We need to show that this combines to correctly sort the entire array.

One detail generally missed is to note that each two-thirds which is sorted is in fact \([2n/3]\) elements. The fact that it is the ceiling and not the floor is significant to there being enough overlap, but we will ignore that in the remainder of this proof.

Let B = (a_1, a_2, ..., a_{2n/3}).

After line 6 (CLM-sort(A,1,2n/3)), the first one-half elements of B (the first one-third elements of A) can never belong in the last one-third elements in the sorted order of A, since we know that the last one-half elements of B are all larger than these.

Therefore the second call to CLM-sort on line 7 involving the last one-half elements of sorted array B and the last one-third of A fixes the last one-third part of A correctly.

Finally, the third call to CLM-sort on line 8 fixes the first two-thirds.

Hence CLM-sort correctly sorts the input data.