Throughout the course, unless otherwise stated, arrays are indexed beginning at 0.

1. The selection sort is usually presented as an iterative (i.e., non-recursive) algorithm. Here is one version:

SelectionSort(array)
    for i = 0 to array length - 1 do
        set minIndex to 0
        for j = i to array length - 1 do
            if array[j] < array[minIndex] then
                set minIndex to j
            end if
        end for
        swap array[i] and array[minIndex]
    end for
end SelectionSort

Here is a recursive version. Code this up in any programming language you choose and test it.

SelectionSort(array, first)
    if first = array length then
        return
    end if
    set minIndex to first
    for i = first to array length - 1 do
        if array[i] < array[minIndex] then
            set minIndex to i
        end if
    end for
    swap array[first] and array[minIndex]
    SelectionSort(array, first+1)
end SelectionSort
2. Implement in the language of your choice the following recursive algorithm for finding the maximum element in an array.

FindMax(array, first, last)
    if first = last then
        return array[first]
    end if
    set middle to (first + last) / 2
    set max1 to FindMax(array, first, middle)
    set max2 to FindMax(array, middle+1, last)
    return the larger of max1 and max2
end FindMax

3. Write a program that is given an array of integers as input and that outputs the length of the longest increasing sequence. A sequence is a series of adjacent elements. For example, in the array [21, 23, 28, 23, 29, 32, 38, 28, 30, 29, 25, 24] the length of the longest increasing sequence is 4. The sequence starts at position 3 (the second occurrence of 23) and ends at position 6 (the value 38). The program should work for any length array.