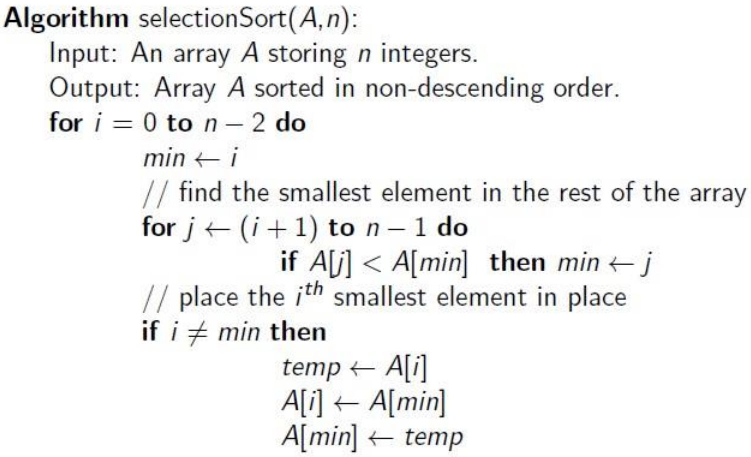
Week 7 CM1210
Exercises

12th March 2019

1 Implementing selection sort

The pseudocode for selection sort was given in the lecture notes for Algorithms Lecture 1, and is reproduced below for your convenience.



```
**Algorithm** selectionSort($A, n$):
    Input: An array $A$ storing $n$ integers.
    Output: Array $A$ sorted in non-descending order.
    **for** $i = 0$ **to** $n - 2$ **do**
        $min \leftarrow i$
        // find the smallest element in the rest of the array
        **for** $j \leftarrow (i + 1)$ **to** $n - 1$ **do**
            **if** $A[j] < A[min]$ **then** $min \leftarrow j$
        // place the $i^{th}$ smallest element in place
        **if** $i \neq min$ **then**
            $temp \leftarrow A[i]$
            $A[i] \leftarrow A[min]$
            $A[min] \leftarrow temp$
```

On Learning Central, download the file SelectionSortTemplate.java. In the main method, there is code to create an `ArrayList`, load it with some words, and call the sorting method.

Your task is to complete the `runSort` method with code to perform the selection sort algorithm, according to the above pseudocode.

2 Testing selection sort

To test that the `runSort` method is both sorting the words as we would expect and using the correct algorithm to do so, print out the contents of the `ArrayList` before and after performing the sorting algorithm. Also, print out some intermediate results so we can see how the sorting algorithm works.

You may want to change the `test String` variable to see how the algorithm works for different inputs.

3 Counting the comparisons/swaps

Print out the word “comparison” every time the algorithm performs a comparison, and “swap” every time it performs a swap. Count the number of comparisons and swaps performed, and output this to the terminal. You may want to start thinking about how different strings influence (or not) the number of comparisons and swaps performed, and how this relates to what you were taught about selection sort in lectures.

4 If you have extra time. . .

- If you’ve completed the above tasks before the end of the lab, modify your code to sort integers instead of Strings.
- You may want to, additionally, try randomly generating some numbers to act as input, instead of hard-coding the numbers like the String was hard-coded into the program.