

Lab #4, Friday 2/9

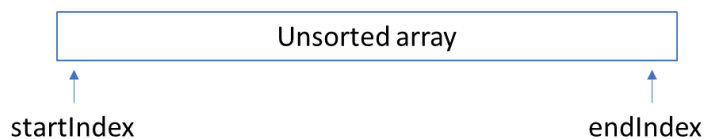
The purpose of this lab is to give you some practice with recursion, arrays, and sorting!

In class we gave an example of how to write a recursive implementation of bubblesort. In this lab you are to write a **recursive** implementation of **selection sort**. You can follow this link, https://www.tutorialspoint.com/data_structures_algorithms/selection_sort_algorithm.htm, to review how selection sort works, and also see selection sort in animated action at <http://sorting.at>. We also went over a non-recursive version in class. You can review the lecture notes or the video.

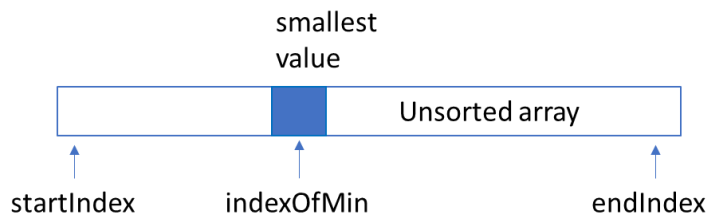
Briefly, in selection sort we find the smallest value in the array and then move it all the way to the left, at index 0. Then we repeat by finding the smallest value in the array starting at index 1, and then move it all the way to index 1. By repeating this we end up sorting the array.

In a recursive formulation, the idea is as follows:

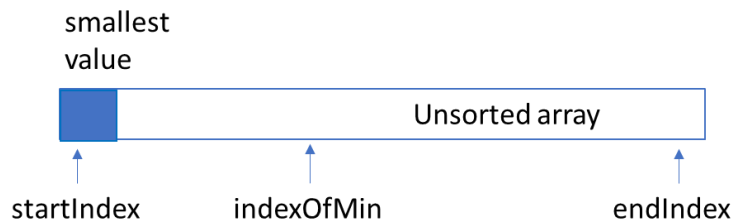
1. Start with an unsorted array. Let `startIndex` be the index of the first element in the array, and `endIndex` by the index of the last element in the array.



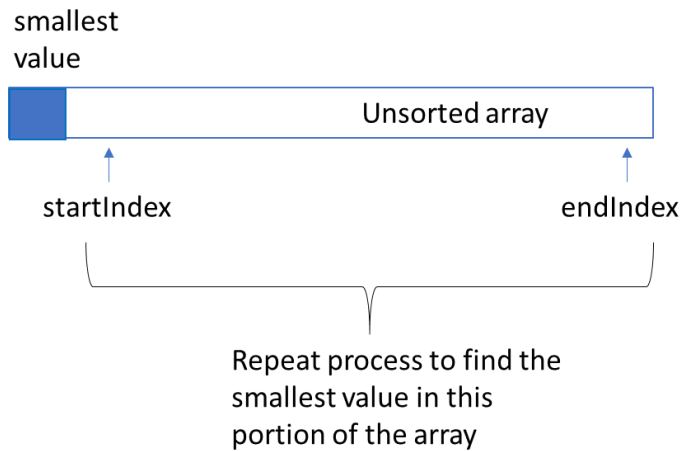
2. Search from `startIndex` to `endIndex` for the smallest value in the array. Let's say that it is found at position `indexOfMin`.



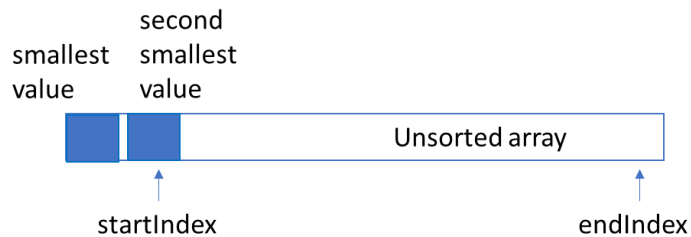
3. Swap the values in the array at `startIndex` and `indexOfMin`. This means that the smallest value in the array is now at `startIndex`.



- Repeat the process, but pretend that the leftmost element isn't there. You can do this by making a recursive call to the `selectionSort` routine, except increase the value in `startIndex` by 1 to bypass the leftmost value:



- When steps 2-3 are repeated on this new portion of the array we end up with the second smallest value next to the smallest value:



- By repeating this process until we reach `endIndex`, we will sort the entire array.

To do: The file **selection.cpp** (linked separately from the calendar page) contains skeleton code for recursive selection sort.

- Complete the `selectionSort` function so it correctly sorts the array
- Write the `swap` function
- Write a main function with a test array, call `selectionSort`, and output the values in the test array to see if the recursive sort is working correctly

Show your working program to the lab assistant or instructor for credit, or email to kjmock@alaska.edu by midnight for full credit (if everything works).