ECE 122: Intro to Programming in Python

Summer 2022

Assignment 5: Sorting and Searching

Deadline: August 19, 2022
Honor Policy: I expect each student to complete this assignment individually.

Description:

For this assignment, you will modify several files associated with Presentations 9 and 10 and make timing measurements. Before starting the assignment, please review the files **searching.py** associated with Presentation 9 and **sorting.py** associated with Presentation 10.

Part 1: Binary Search and Linear Search

This purpose of this exercise is to make small changes to the provided linear and binary search code to allow for searches over a portion of a provided list (rather than the whole list). You will modify the **sequentialSearch** function in *searching.py* to accept a parameter **percentage.** This parameter indicates the percentage of the list, starting from index 0, that will be searched. At the end of the search, the list will be modified to only include the values that have not previously been searched. Note that you should not create a new list to make this modification. Consider using the construct *alist[:].* Hint: Modify the while loop so that a fixed number of values are searched. The whole list can be searched if percentage equals 100.

You will also modify **binSearch** to accept a parameter *iteration*. This parameter indicates the number of iterations of the binary search that will be performed. If this number is small, the entire list may not be searched. At the end of the search, modify *mylist* to only include those values that have not previously been searched.

Note that it is not necessary to modify *part1.py* for this task. Only *searching.py* should be modified.

Example:

Let’s say the list includes the values [1,2,3,4,5,6,7,8, 9,10], the search value is 6 and the number of iterations for binary search is 2. For the first iteration, the lower value is 5 and upper value is 10. For the second iteration the lower value is 5 and upper value is 7, so the updated list is now [5,6,7]. If we were to do a linear search now of 60 percent, we would search [5,6] and find 6.

What to submit for this task: Modified *searching.py* and answers to the following questions in a report.

1. **When does linear search outperform binary search in time? Can it ever be faster?**
2. **Are there any benefits to doing linear search and binary search together, as shown in this task?**

Part 2: Binary Search Return Index

For this part, you will modify the **binSearch** code in the original *searching.py* to return the index where the first occurrence of the search value is found rather than True or False. If the search value is not found, -1 is returned.

What to submit for this task: Modified **binSearch** function in a file *part2.py*.

Part 3: Reverse Sorting

In this part, you will modify the selection sort algorithm to sort values in a list in descending order (biggest to smallest) rather than ascending order. No other changes to the code are needed.

What to submit for this task: Modified **reverse_selection_sort** function in *part3.py* and answers to the following questions (use code in the main part of *part3.py* to create answers to the questions)

Questions:

1. Does a reverse order sort affect the sorting algorithm's runtime?

2. What if the lists were sorted or almost sorted? Would it impact selection sort performance?

3. As the list sizes double, what happens to the runtimes of the sorts?