Lab Report 2

CS 2302 Data Structures

Spring 2019

Miriam Ramirez

**Introduction**

This lab consisted of two parts. The first part requires us to create three functions where we are to implement two sorting algorithms: bubble sort and quick sort. Two of the functions will use the bubble and quicks sort algorithms recursively to return numbers from smallest to largest given a list. The third function will also implement the quick sort method, but will only call itself recursively once.

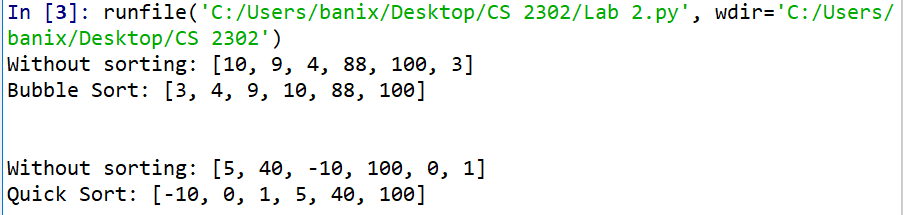
**Proposed Solution Design and Implementation**

In order for me to be able to come up with a solution for part one of the lab, I had to first understand how the required sorting methods worked: bubble and quick. Both sorting methods were taught in Elementary Data Structures so a refresher was simply necessary, for bubble sort at least, quick sort still takes me a while to grasp. In part one of the lab implementing those two sorting methods was the easiest thing I could do, so I began with it.

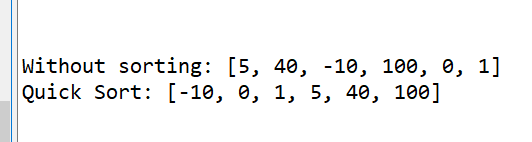
**Experimental Results**

I created three lists, each with different numbers and all out of order in order to test the three sorting methods I created and to make sure they ran properly.

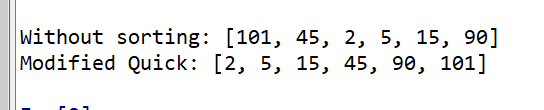
The bubble sort method:



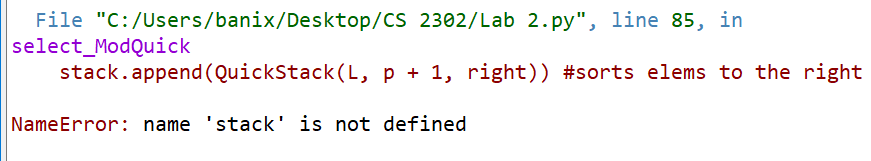
The quick sort method:

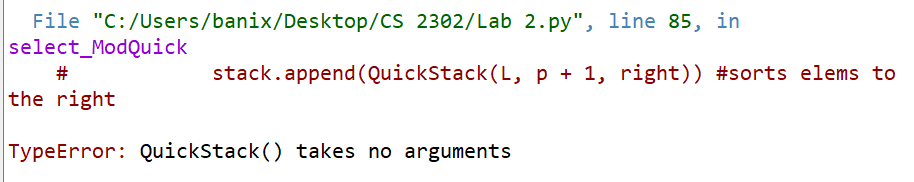


The modified quick sort method:



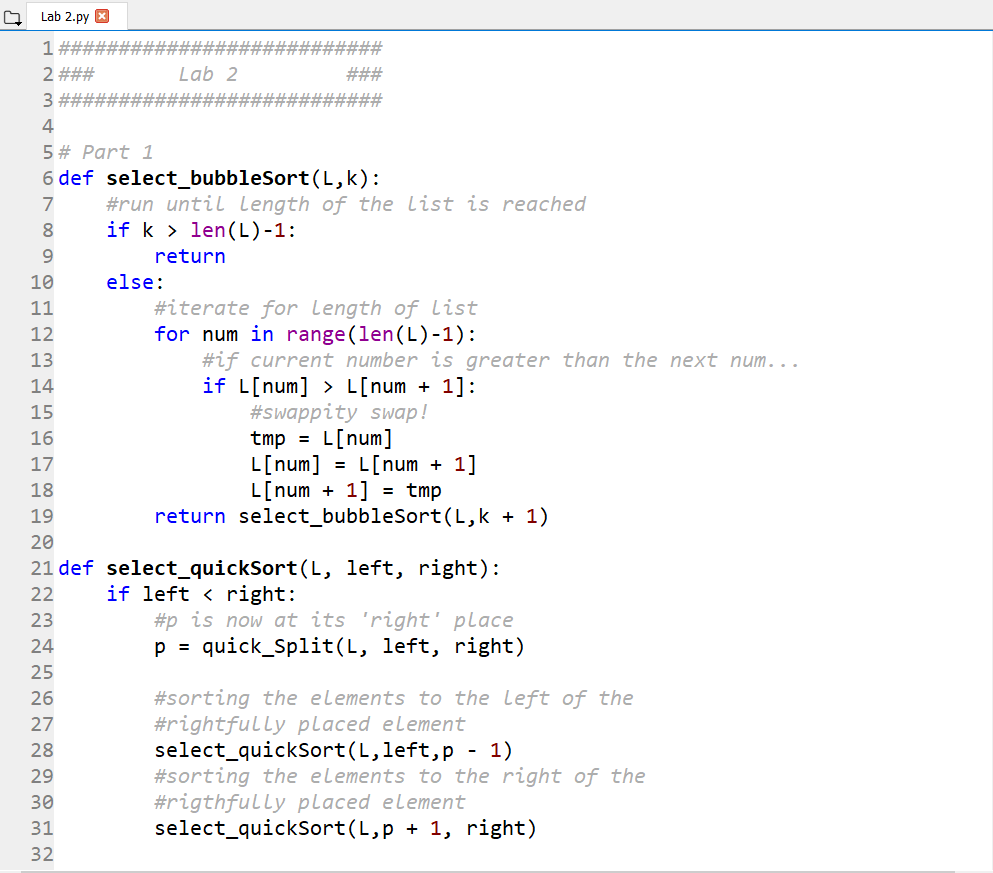
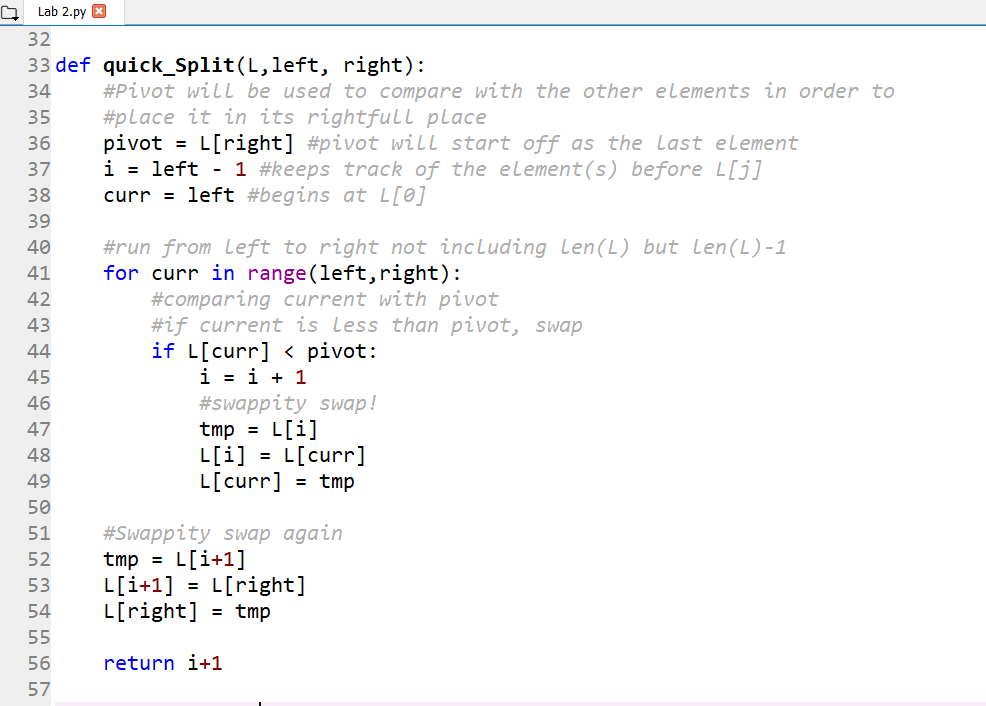
When I reached part two, I kept getting this error when implementing the stack onto the select\_modifiedQuick() method:





**Conclusions**

I learned the running times of each of the algorithms performing. I had already learned bubble sort and quick sort in Elementary Data Structures, but this lab helped refresh what I had already forgotten. Also, I learned that it is possible to create a quick sort with one recursive call and although I didn’t get to it, quicksort can also be implemented using stacks and using simply a while loop.

**Appendix**

