Computer Science I — CSCI-141
SelectionSort — Homework

09/28/2015

1 Problem

Compose a program that reads numbers from a file, uses selectionsort to sort the numbers, and has comments answering the questions in this assignment.

The *selectionsort* algorithm involves the following invariant. The sequence of values can be cut into two sub-sequences such that the first sub-sequence is in order *and* all the elements are less than or equal to the elements of the second sub-sequence. To satisfy this invariant, initially, the first sub-sequence is empty. To extend the first sub-sequence while preserving the invariant, we find the smallest value of the second sub-sequence and move it to the end of the first sub-sequence.

Here is an example of selectionsorting:

	|	$45, 7, 5, 24, 12, 1$
1	|	$7, 5, 24, 12, 45$
$1, 5$	|	$7, 24, 12, 45$
$1, 5, 7$	|	$24, 12, 45$
$1, 5, 7, 12$	|	$24, 45$
$1, 5, 7, 12, 24$	|	45
$1, 5, 7, 12, 24, 45$	|	

The pseudo-code[1] below suggests how the selectionsort algorithm might be implemented. The function `findMinFrom` is assumed to return the index of the smallest value in the list for the specified range.

```
define selectionSort( lst )
    for mark in 0 .. len(lst) - 1
        swap(lst, mark, findMinFrom(lst, mark))
```

1.1 Constraints

- Use iteration to implement the algorithm (i.e., use *for* or *while* loops).
- Sort the list of numbers in-place; create no extra lists.
- You must implement the specified algorithm and no other.
- You may not use any built-in Python sort functions.

[1] Pseudo-code is a form of description that looks like some programming language but is not an actual language. This permits the clear description of an algorithm with less concern for syntactic correctness.

1.2 Program Tasks [90%]

Your solution must do the following:

1. Prompt for an input file name. The file will consist of integer values, one per line.
2. Open the file, read the file, and store the data in a list.
3. Print the list.
4. Call your selectionSort procedure passing the list.
5. Reprint the list, which is sorted.

Develop and use your own input files to test your solution.

1.3 Questions To Answer [10%]

Answer the questions below after you implement and test your program. **Write and number your answers to the questions in the docstring that contains your name at the start of your program file.**

1. In what kind of test case does insertionsort perform better than selectionsort? Clearly describe the test case.
2. Why does selectionsort perform worse than insertionsort in that test case?

You may want to copy and customize the **insertionsort program** to make it work with files, and run the insertionsort code with the same file inputs.

1.4 Submission

Submit your solution in a file named `selectionsort.py` to the myCourses dropbox for this assignment.