CS2040S: Data Structures and Algorithms

Exercises/Optional Problems for Week 2

For: January 21, 2020

Problem 1. String Sorting but actually not String Sorting

Sorting is a really fun way to order things. In particular sorting strings is particularly interesting since there are many ways to order them. For example there is the *lexicographic* ordering which is the default String comparison that Java does. In lexicographic ordering, given two strings $a = a_1a_2a_3 \dots a_m$ and $b = b_1b_2b_3 \dots b_n$, we say that $a \prec_{lex} b$ if any of the following are satisfied.

1. $a = \varepsilon$
2. If $m \leq n$ and $a_i \leq b_i$ for all $i \in [1 \dots m]$
3. If for some integer k where $k \leq m, n$, and $k \geq 1$, for all $i \in [1 \dots k-1]$ $a_i = b_i$ and $a_k \neq b_k$ and in fact $a_k < b_k$ in terms of character comparsion.

However, sorting the normal way is boring. Given a class list, you are going to label the tables with the first two letters of each student's name. You would also like these two letter names to be in running order (i.e. in sorted order).

In addition to this, you would like to minimise the number of swaps within the class list that you require to do. Hence you would like your sorting algorithm to be **stable**. Recall that a sorting algorithm is stable if given two objects a and b, if $a = b$ based on the comparison used by the sorting algorithm and a appears before b in the original list, then a will appear before b in the sorted list.

You are given the class file `Sorter.java`, as shown below:

```
public class Sorter {

    public static void sortStrings(String[] arr) {
        // TODO: implement your sorting function here
    }
}
```

The function `sortStrings(String[] arr)` takes in an array of strings and performs a sort **in place** on the given array. You may refer to `SorterTest.java` to see how this method is called.

You may find implementing the following function to be helpful as well

```
    public static boolean isGreaterThan(String str1, String str2) {
        //your implementation here
        return false
    }
```

which takes in two strings and outputs whether `str1` $\prec$ `str2`. based on what ordering you wish $\prec$ to represent. Use the tests in `SorterTest.java` to test your implementation against sample tests. When you are done, only upload the completed file `Sorter.java` to Coursemology.