Lab Report 01

Nathan Bickel

Problem

The task was to build a helper class that would offer static methods to sort a list in ascending order based on the number of vowels in the word, number of consonants in the word, and simply number of letters in the word. The methods were to be written to take in an unsorted String array and return a String array that is sorted.

Solution

The first thing I did was to create a helper method that takes in three strings as parameters and returns a boolean. The boolean evaluates to true if there are more characters in the first string than in the second string that are contained in the third string. For example, if the first string was "adieu", the second string was "fitness", and the third string were a string called vowels initialized as "aeiouy", the method would return true because the first string has four characters contained in the third string of vowels and the second string has two fewer (2) characters contained in the third string. This was implemented by initializing two counts as 0 for the first two strings, iterating through each character in each string, and increasing the corresponding count if that letter was contained anywhere in the third string.

Then, for the vowel sort and consonant sort, I could use the same helper method with different third strings, and although I used the .length() method for the number of characters method to save on computation, I could have used the same helper method with a string containing all letters. For each method, I created a new array that had the same words as the parameter, used bubble sort to sort by the desired constraint, and returned the array. To bubble sort, I initialized a boolean representing whether any swaps had been made, and then each time that value was set to true, I iterated through the list and swapped any pairs that were out of order using the helper method. Once no swaps were made during a pass, it was sorted, and bubble sort was complete.

Implementation Problems Encountered

I had several implementation problems that I had to debug throughout developing the helper class.

With bubble sort, I got confused with what state it should be initialized in and when to switch it, and it took me a little while to realize that when I was running the program the loop was only running once.

Another mistake I made was I initially tried to use bubble sort on the parameter String array itself rather than a copy of it. This was causing the array to be changed around, and thus the original word list was being changed and being printed out sorted by the last method that ran on it. Once I realized that, I tried to fix it by making a copy, but I created an array and simply set it equal to the parameter. They were pointing to the same memory address, so this resulted in exactly the same problem. I realized that I needed to create a new array at a different address and repopulate it with the values from the parameter using the copyOf() method (or alternatively using a for loop), which fixed the issue.

I also had a few minor issues. I used the substring method to check which character existed at each index because .contains() takes a String parameter rather than a char parameter, so I couldn't use charAt(). However, I initially wasn't indexing the last character because I forgot that the second index bound is exclusive, so that threw me off. Additionally, I wasn't sure whether or not to include "y" as a vowel or a consonant, but I decided to include it as a vowel. Finally, I somehow managed to include "i" in my string of consonants and didn't catch it after half an hour of debugging, so that had to be corrected.

Lab Report Questions

1. The primary advantage of using an array is the ability to have instantaneous random access. Because the array represents contiguous blocks in memory, each index can be accessed essentially instantaneously.
2. The main disadvantage of an array is that it cannot be resized. The size must be set when it is declared, so if the size ever changes, a new array must be created and the data must be moved into the new array. Thus, elements can not be added or removed easily.