**Lab 6 – Sorting and Big-O**

**CSC 3302**

|  |  |
| --- | --- |
|  |  |
| **Introduction:** | Runtime complexity is an issue when determining which algorithms we should use. Some algorithms are easy to implement but run too slowly to be useful. Knowing the advantages and disadvantages of different algorithms helps in determining the best solution to a problem. |
|  |  |
| **Description:** | You are to implement two different sorting algorithms and count the number of steps involved with the sort routine (the comparisons and moving positions). Every time a comparison or move is made you should count it.  You have been given a data file with text. You must read through the file to determine the number of words in it, then create an array big enough to hold the values. Once the array is built you need to reread the file to store the strings into the array.  You must implement **Insertion Sort** and **Quick Sort**. These sorts must take an array of strings and sort them in ascending order. When sorted you must display each UNIQUE value and the number of times it appears in the array. At the end you must print out which sort was used and how many steps it took to complete.  You must keep asking the user which sort to use until the Exit symbol is entered. Remember that passing an array to a method will modify the original array. You must copy the array so you do not lose the order of the original and just sort the copy. |
|  |  |
| **Program:** | You must create a program called **p6.java**. Everything can be done inside of this file if done within the same class. If you wish to create additional classes to help, you must have them in separate .java files. |
|  |  |
| **Input:** | There is an input file called **p6.dat** were each line contains a name. |

|  |  |
| --- | --- |
| **Output:** | (S)election Sort  (Q)uick Sort  (E)xit  Enter choice: Q  A : 56 frequency.  AM : 2 frequency.  AND : 25 frequency.  ANYWHERE! : 3 frequency.  ANYWHERE. : 5 frequency.  ARE : 1 frequency.  ARE. : 1 frequency.  BE! : 2 frequency.  BE, : 1 frequency.  BE. : 1 frequency.  BOAT. : 2 frequency.  BOAT? : 1 frequency.  BOX. : 6 frequency.  BOX? : 1 frequency.  CAR! : 3 frequency.  CAR. : 3 frequency.  CAR? : 1 frequency.  COULD : 14 frequency.  DARK! : 2 frequency.  DARK. : 3 frequency.  DARK? : 2 frequency.  DO : 36 frequency.  DO! : 1 frequency.  EAT : 23 frequency.  EGGS : 11 frequency.  FOX. : 6 frequency.  FOX? : 1 frequency.  GOAT! : 1 frequency.  GOAT. : 1 frequency.  GOAT... : 1 frequency.  GOAT? : 1 frequency.  GOOD, : 2 frequency.  GREEM : 1 frequency.  GREEN : 10 frequency.  HAM! : 3 frequency.  HAM. : 6 frequency.  HAM? : 2 frequency.  HERE : 11 frequency.  HOUSE. : 7 frequency.  HOUSE? : 1 frequency.  I : 69 frequency.  IF : 1 frequency.  IN : 40 frequency.  LET : 4 frequency.  LIKE : 44 frequency.  MAY : 2 frequency.  MAY, : 1 frequency.  MAY. : 1 frequency.  ME : 4 frequency.  MOUSE. : 6 frequency.  MOUSE? : 1 frequency.  NOT : 67 frequency.  NOT, : 15 frequency.  ON : 6 frequency.  OR : 8 frequency.  RAIN. : 3 frequency.  RAIN? : 1 frequency.  SAM! : 2 frequency.  SAM, : 1 frequency.  SAM-I-AM! : 4 frequency.  SAM-I-AM. : 9 frequency.  SAM. : 2 frequency.  SAY! : 3 frequency.  SAY. : 2 frequency.  SEE! : 1 frequency.  SEE. : 3 frequency.  SO : 5 frequency.  THANK : 2 frequency.  THAT : 3 frequency.  THE : 11 frequency.  THEM : 40 frequency.  THEM! : 4 frequency.  THEM, : 10 frequency.  THEM,SAM-I-AM. : 1 frequency.  THEM. : 3 frequency.  THEN : 1 frequency.  THERE. : 8 frequency.  THERE? : 1 frequency.  THEY : 2 frequency.  TRAIN! : 5 frequency.  TRAIN. : 2 frequency.  TRAIN? : 1 frequency.  TREE! : 3 frequency.  TREE. : 3 frequency.  TRY : 4 frequency.  WILL : 18 frequency.  WITH : 18 frequency.  WOULD : 27 frequency.  YOU : 23 frequency.  YOU! : 1 frequency.  YOU, : 8 frequency.  YOU? : 2 frequency.  Quicksort finished in 12000 steps. |
|  |  |
| **Hints:** | Note that depending on implementation the number of steps you will have is different from others.  Use appropriate software design techniques, and implement the class methods with Java constructs for I/O, declarations, and calculations.  Build your program in steps (i.e., get the input and output working, then add the functions, etc.). Emphasize functionality first, then add the advanced features. Work on this program over time, DO NOT wait until the last minute (i.e., the day before it is due) to start! |
|  |  |
| **Data:** | On the server, you will need to use the following absolute path and append the filename to it. "/home/courses/csci3302-002/datafiles/" |
|  |  |
|  | Remember that you must pass the data file name in as a command line argument. |
|  |  |
|  |  |