CS146 Homework 4

These problems will help you understand the searching algorithms.

Submitting Information:

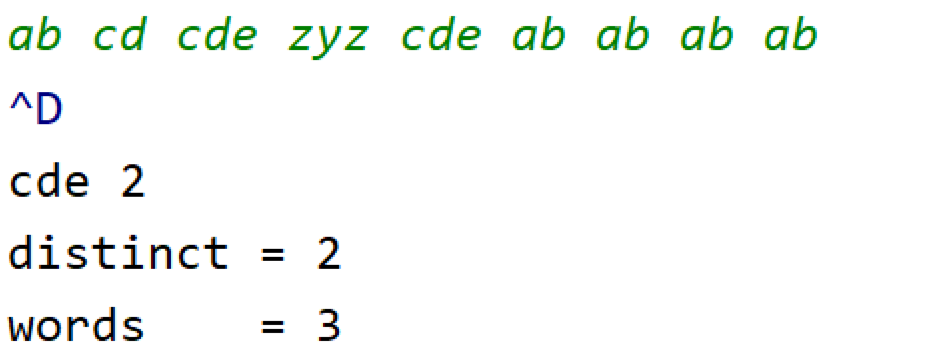
- DO NOT post your code on Piazza
- Use the code I provided for each problem. DO NOT delete any function
- Submit your work on Canvas.
- DO NOT change the name of .java files.
- Use javac-algs4 to compile and java-algs4 to run the code
- Put all the .java files in one zip file and name it <last_Name><first_initial>.zip
- The deadline is Friday, April 21st at 11:59PM
- There should not be compilation error in your code and It should produce the expected output

Problem 1 (15 points). *(Array-based Symbol Table)* Develop a symbol-table implementation ArrayST that uses an (unordered) array as the underlying data structure to implement the basic symbol-table API (p. 363).

```
$ java ArrayST Pluto
Mercury Venus Earth Mars Jupiter Saturn Uranus Neptune Pluto
<ctrl-d>
Mercury 1
Venus 2
Earth 3
Mars 4
Jupiter 5
Saturn 6
Uranus 7
Neptune 8
```

Problem 2. (15 points) *(Frequency Counter)* Modify FrequencyCounter from the text to use ArrayST from above and also to print all of the values having the highest frequency of occurrence, instead of just one.

Output for Program when running with argument 2:



```
ab cd cde zyz cde ab ab ab ab
^D
cde 2
distinct = 2
words    = 3
```

Problem 3. (15 points) *(Average GPA)* Write an ArrayST client called AvgGPA that creates a symbol table mapping letter grades to numerical scores, as in the table below, then reads from standard input a list of letter grades and computes and prints the average GPA (the average of the numbers corresponding to the grades).

A+	A	A-	B+	B	B-	C+	C	C-	D	F
4.33	4.0	3.67	3.33	3.0	2.67	2.33	2.0	1.67	1.0	0.0

```
$ java AvgGPA
B A- A A- B A- A- B A+ B A+ A- B- B B+ B+ A C A+ F
<ctrl-d>
3.2835
```

Problem 4. (20 points) Implement a code which takes in-order traversal and pre-order traversal as input and output's the post order traversal.[Hint: Creating a BST from in order and preorder traversal can help to find postorder traversal]

```
$ java PostOrder
4,2,5,1,3
1,2,4,5,3
<ctrl - d>
4,5,2,3,1
```

Problem 5. (20 points) (Tree Traversal) Implement the methods preOrder(), inOrder(), postOrder(), and levelOrder() in TreeTraversal that return the an iterable object containing nodes of a binary tree traversed in pre-, in-, post-, and level-order, respectively.

```
$ java TreeTraversal
F B G A D I C E H
<ctrl-d>
Pre-order:   F B A D C E G I H
In-order:    A B C D E F G H I
Post-order:  A C E D B H I G F
Level-order: F B G A D I C E H
```

Acknowledgement: This homework is adaption of work from Swami Iyer.