HW05

Comparing Library Works with Hash Tables

The Idea

- Two different authors: Shakespeare and Bacon
- One very long text from each **(DO NOT PRINT)**
- We give you file input routines to read the words from each into two separate arrays of strings in order from the text.
- Now you are going to create two hash tables, one for each author that will keep a count of how many times each word appears in that author's work.

The Data: REALLY BIG

- Hamlet by Shakespear

Project Gutenberg Etext of Hamlet by Shakespeare
PG has multiple editions of William Shakespeare's Complete Works
HAMLET, PRINCE OF DENMARK
by William Shakespeare
PERSONS REPRESENTED.
Claudius, King of Denmark.
Hamlet, Son to the former, and Nephew to the present King.
Polonius, Lord Chamberlain.
Horatio, Friend to Hamlet.
Laertes, Son to Polonius.
Voltimand, Courtier.
Cornelius, Courtier.
Rosencrantz, Courtier.
Guildenstern, Courtier.
Osric, Courtier.
A Gentleman, Courtier.
A Priest. ....

More Data: ALSO BIG

- Bacon Essays

The Project Gutenberg EBook of Essays, by Francis Bacon
THE ESSAYS OR COUNSELS, CIVIL AND MORAL,
OF FRANCIS Ld. VERULAM VISCOUNT ST. ALBANS
By Francis Bacon
THE ESSAYS
- Of Truth
- Of Death
- Of Unity in Religion
- Of Revenge
- Of Adversity
- Of Simulation and Dissimulation
- Of Parents and Children
- Of Marriage and Single Life
- Of Envy
- Of Love
- Of Great Place
- Of Boldness
- ...

Hash Tables

entry	count
Hamlet	3457
king	895
shall	4000
...	

entry	count
truth	650
parents	346
shall	24
...	

Total counts: 8352 1020

Frequencies: (count/arraysize)

Hamlet	.4139		truth	.6373
king	.1072		parents	.3392
shall	.4789		shall	.235

Squared Error = $(.4139-0)^2 + (.1072-0)^2 + (.4789-.235)^2$
$+ ((.6373-0)^2 + (.3392-0)^2$

from going through first table and checking second

from going through second table and checking first (don't count shall again)

General Distance Metric

- For each element e with frequency f in Shakespeare, but not in Bacon, add f^2 to the error.
- For each element e with frequency g in Bacon, but not in Shakespeare, add g^2 to the error.
- For each element e with nonzero frequency f in Shakespeare and nonzero frequency g in Bacon, add $(f-g)^2$ to the error.

Hash Table Implementations

- You will do this twice, 2 separate programs:

1. use chaining
2. use quadratic probing

- You will be given starter code and write a number of functions yourself
- For converting a character string to an integer, you may use Java's HashCode method.
- But you can write your own for extra credit.

Main Functions to Write

- Constructors for hash tables
- insert(String keyToAdd): adds keyToAdd to table if not there, setting count to 1, else adds 1 to count
- findCount(String keyToFind): returns the count
- getNextKey(): iterator that returns the next key in a hash table, used when going through the table to compute the distance metric

A Few More Details

- Insert will be different for the chaining and for the probing.
- You will likely write multiple small helper functions as you do this.
- Test.java has a bunch of TODOs.
- At the end you will print

1. the TOTAL error for the two tables
2. the word with the highest frequency difference