

CMSI 282 Problem Set #3
Due February 26, 2009

1) Make `public class Trie`; an instance of this class is a single trie (retrieval tree) for storing a multiset of words. It should implement at least these methods:

- `public Trie (char[] legalChars);`

This constructs and initializes a trie whose words may contain (only) the characters in the array `legalChars`.

- `public boolean isLegal (char ch);`

This method returns `true` iff `ch` is one of this trie's allowable characters.

- `public void insert (String word);`

This method adds `word` to the trie or, if already present, increments the frequency associated with `word`.

- `int frequency (String word);`

This method returns the frequency associated with `word`.

- `String toString ();`

This method returns a string that represents the trie and is suitable for printing. Specifically, the string should be an alphabetical listing of all words in the trie, together with each word's frequency, and each word-frequency pair delimited by a newline character.

2) Make a trie-based program, `Concordance.java`, that outputs, in dictionary order, all of the words that appear in a given text file, along with their frequencies. A typical invocation of your program might look like this:

```
java Concordance -sa < GreatExpectations.txt
```

The `"-s"` option indicates that the program should be case-sensitive, i.e., that `"cat"` and `"Cat"` and `"cAT"` should be considered as different strings; by default, the program should treat upper- and lower-case letters the same. Here are some additional guidelines:

- a) When forming words, only letters and digits count. All other characters serve as delimiters, except as noted below.

- b) Ignore all hyphens, so, e.g., “ca-t-s” and “c—at-s-“ are equivalent.
- c) By default, ignore all apostrophes; however, your program should count them if the “-a” option is on.
- d) Generally, ends-of-lines should be treated as delimiters; an exception is when the end-of-line happens to coincide with a hyphenated word that is being continued on the next line.
- e) Clearly state all other assumptions, command-line options, etc.
- f) You should work in pairs for this assignment. In the *pair programming* paradigm, you do not- and, for this assignment, you *may* not- simply partition the problem and work independently; rather, both people work *closely* to ensure proper design and coding. As a prerequisite, you and your partner should read http://en.wikipedia.org/wiki/Pair_programming .