CS 240
Week 2

Introduction to Project 2

- Command Line Syntax

  ```
  java Spell word
  ```

  -- source file hard wired into program
- Run Demo

Basic Algorithm

- Look up word in dictionary
  - Comparison is case insensitive

-
  ```
  If found
      List the word
  Else if a similar word is found
      List the most similar word
  Else
      Indicate no word found
  ```
- See project description for syntax
- The input word must consist of 1 or more letters

Dictionary Must Be Represented as Trie

- A tree like structure defined recursively
  - Root node
  - Internal nodes (sub-Tries)
    - Leaf nodes have no sub-Tries
- Each node contains
  - A count
    - A value > 0 represents the number times a word in the dictionary (represented by a path through the tree to the current node) appears in the dictionary.
  - 26 Sub-Tries having the values ‘a’..’z’
- The Trie (or root of the Trie) contains
  - The number of words in the dictionary
  - The number of nodes in the dictionary

Trie Operations

- Constructor
- Methods
  - add(String word)
  - find(String word); may return null
  - int getWordCount()
  - Int getNodeCount()

What Does It Mean for a Word to Be Similar to Another Word

- Edit Distance
  - Deletion distance
  - Transposition distance
  - Alteration distance
  - Insertion distance
- Edit Distance 1 and 2

Making the Output Deterministic

- One word is “more similar” than another word

  Based on the following priorities

  1. A word that is has an edit distance of one has a higher priority than one that is has an edit distance of 2
  2. If both words have the same edit distance then select the one that appears more frequently in the dictionary
  3. If two words appear equally often in the dictionary then pick the one the is alphabetically first.

Additional String Operations

- s.toLowerCase()
- s.toUpperCase()
- StringBuilder
  - Building strings using “+” is expensive
  - StringBuilder strBuilder = new StringBuilder()
    - strBuilder.append(String)
      - Can be anything that has a toString method or atomic type
    - strBuilder.delete(int, int)
    - strBuilder.setCharAt(int,char)
- Code Examples

Interfaces

- Interaction contract
  - At any place in a java program you an have variables of the interface type
  - Object in variables must be instance of an implementing class
- Keyword “implements”
- Like .h file in concept (conceptually similar to abstract class)
- Contains
  - Constants (static and final)
  - Method signatures
- You must provide an implementing class

Interface Example

- StudentListInterface.java: The interface
  - ArrayStudentList.java: implements StudentListInterface.java
  - ArrayListStudentList.java: implements StudentListInterface.java
- TestingStudentListInterface.java

Interesting Addition of For Loops

- for(String s = “”; s != null; s = f.readLine()) …
- Use of finally
- Example Code
  - testFile – put in same directory as Example Code