

## CSCI Program Seven

Assigned Tuesday Nov. 5, 2013

Due Tuesday Nov. 12 by midnight

You may base your solution on course notes and/or your textbook. All computer code that you submit must represent your own effort and understanding of the code. You may speak to classmates - only exchanging human language and not code, or Ed Pharr, the course assistant.

Automated systems that monitor electronic messages such as e-mail often need to rank the words that appear in a given message to identify which words in the message are the most significant. For example, a spam filter would want to identify those words of a message that appear in a dictionary of likely spam words such as "Viagra" or "lottery". A government security agency, may want to identify words in a message that appear in a dictionary of known likely criminal or terrorist activity such as "Al Qaida" or "sabotage". These systems usually rely on a pre-built dictionary that for each word, gives an estimated risk rank, with higher risk rank values denoting words that are more likely to be of interest.

You will be provided with Java code to load a risky words dictionary into a Dictionary class that provides these public methods:

**public boolean isWord(String W)** – returns true if the given word (case insensitive) is in the dictionary; else, returns false.

**public double getRank(String W)** – returns risk rank of given word as a type double value between 0.0 and 1.0, where a higher value denotes a greater risk. If a word is not found, then it returns a risk of 0.0.

**Implement the following classes – RankedWord, WordPriorityQueue, and Prog7.**

### **public class RankedWord**

Represents one word and its risk priority such as word: "sabotage" risk: 1.0.

Private instance variables:

word – String for a given word

rank – Type double value between 0.0 and 1.0, inclusive, with higher values denote greater significance.

Public methods:

RankedWord() – default constructor initialize to empty string and rank of 0.0.

RankedWord(String S, double R) – sets word to S and sets rank to S

String getWord() – returns the word as a String

double getRank() – returns the rank assumed to be between 0.0 to 1.0, inclusive.

### **public class WordPriorityQueue**

Implements a priority queue of RankedWords by using a MAX HEAP.

Private instance variables:

fixed-size array of to hold up to 1023 RankedWord objects.

integer count of number of words currently stored in the priority queue

Public methods:

WordPriorityQueue() – default constructor initializes to empty and creates fixed size array to hold up to 1023 RankedWord objects. Count of words stored is 0.

boolean isFull() – returns true if capacity has been reached; else, false.

boolean isEmpty() – returns true if no words are yet stored.

boolean insert(String W) – if not full, then insert given word into the MAX HEAP and return true; else, do nothing and return false. Duplicate words can be inserted.

RankedWord remove() – if not empty, remove and return the highest remaining RankedWord from the priority queue; else, return null if empty.

### **public class Prog7**

Your main method will perform the following steps:

Call on provided code to load dictionary of risk-ranked words.

Call on provided code to load a list of words found in a given text file message.

For each word in the message do

    Retrieve the RankedWord from the risk-ranked words dictionary

    Insert that RankedWord into the priority queue

Print the top five most risky words found in the message.

Loop 1 to 5

    Remove RankedWord from priority queue

    Print word and its risk rank

### **Grading Criteria**

10/10	RankedWord Class
10/10	WordPriorityQueue instance variables/isEmpty/isFull
25/25	WordPriorityQueue insert
30/30	WordPriorityQueue remove
15/15	Prog7 main driver
10/10	Method header comments for RankedWord and WordPriorityQueue

### **Files to Submit to OAKS Drop Box**

RankedWord.java

WordPriorityQueue.java

Prog7.java

**FYI – Learn more about how basic spam filtering works.**

[http://www.process.com/precisemail/bayesian\\_example.htm](http://www.process.com/precisemail/bayesian_example.htm)