CMPSCI 187 Discussion #3: Sorting a StringBag with Stacks
Individual Handout

David Mix Barrington
26 September 2012

Our goals here are to write a program using external code and an external file, and to demonstrate the use of stacks. What we need from outside is:

- Tom's `StringBag` class from Project #1, modified by me to include one more method,
- The `LinkedStack` class (and necessary associated classes) from DJW's code base, *or* just `java.util.Stack`, and
- A file `words.dat`, containing the 5757 five-letter words in English, according to Donald Knuth. These are all in lower case,have a line break after each,and are *not* ordered alphabetically.

Our idea to sort the words is to use two stacks, `left` and `right`. We want to insert words into the two stacks, maintaining the invariants that:

- The words in `left` are in alphabetical order with the first word on the bottom.
- Every word in `left` comes before every word in `right`.
- The words in `right` are in alphabetical order with the first word on the top.

To maintain these invariants while inserting a new word, you need the new word to be between the word on the tops of the two stacks. To get that to happen, you need methods to shift the top word of `left` to be the top word of `right`, and vice versa. Note that the syntax for stack operations differs between the two implementations.

Your task is to write a program that will successively:

1. Create a `StringBag` object and insert each string from `words.dat` using the method `loadFromFile` in the `StringBag` class, e.g., `sb.loadFromFile("words.dat");` where `sb` is the `StringBag`.
2. Create two `LinkedStack<String>` or `Stack<String>` objects `left` and `right`.
3. Remove each string from your `StringBag` using the zero-parameter `remove( )` method, and place it into one of the stacks so that the two stacks hold their strings in order.
4. Insert the strings back into your `StringBag` in such a way that they are now in alphabetical order.
5. With five `remove(k)` operations, determine the 1000th, 2000th, 3000th, 4000th, and 5000th word in the list in alphabetical order (in locations 999, 1999, 2999, 3999, and 4999). Record these on your response sheet.