Lab Exercise 10
Using Maps

In this lab we introduce Maps. Maps are introduced in section 5.7 of the textbook. Maps are similar to collections except that each element is a key/value pair. Each pair may be retrieved by its key, so that we can say that a key maps to a value. Our map is implemented as an array. Entries are sorted by their keys. Words are stored as keys and word counts are stored as values. Sorted entries allow for binary search over the entries.

Using Eclipse, import “Lab10” in the following directory:

/afs/cs.uwm.edu/users/classes/cs351/401/pantherid/git/lab10.git

1 Binary Search

Most of the SortedArrayMap has already been implemented for you. You must complete the private binarySearch method, which returns the position index of the entry or null if it doesn’t exist. Below is an example:

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;and&quot;</td>
<td>&quot;deed&quot;</td>
<td>&quot;fee&quot;</td>
<td>&quot;hello&quot;</td>
<td>&quot;jack&quot;</td>
<td>&quot;jam&quot;</td>
<td>&quot;look&quot;</td>
<td>&quot;moon&quot;</td>
</tr>
</tbody>
</table>

searching for "hello"

lo: 0
hi: 8
mid: 4

"hello" < "jack"
search left side

lo: 0
hi: 4
mid: 2

"hello" > "fee"
search right side

lo: 3
hi: 4
mid: 3

"hello" == "hello"

2 Word Counter

Finally, in the WordCounter class, complete the implementation for countString, getCount, and getContents by using the map’s methods.

Once you have completed the above and passed the supplied tests, see your TA to receive credit for this lab.