Kenneth Marenco and kem78

Run WordGramBenchmark for wordgrams of size 2-10 and record
the number of WordGram values/objects that occur more than
once as reported by the runs. For example, with WSIZE = 2,
which generates 2-grams, the output of benchmark and benchmarkShift
each indicates that the total # wordgrams generated is 177,634
and that the # unique wordgrams is 117,181

This means there are 177,634 - 117,181 = 60,453 WordGram values that
occur more than once. Find these same numbers/values for other orders
of k and complete the table below for different k-grams/different
values of WSIZE

WSIZE	# duplicates
2	60,453
3	10,756
4	1,987
5	667
6	362
7	226
8	151
9	105
10	73

=====

Explain in your own words the conceptual differences between
the benchmark and benchmarkShift methods.

I believe that the main differences are that the benchmark method starts with scanning
the entire txt file and then using that amount of data through each process (i.e. adding
to the ArrayList, converting the List to a String array, and then adding to the set).
BenchmarkShift is quicker because it initially starts a String array the size of WSIZE, and
then sets up the Set with the small WordGram. Now, the scanning process can be done
immediately with smaller batches using the .shiftAdd() method which allows smaller therefore
quicker batches to be added to the set directly.

Answer these questions:

(1) Why the results of these methods should be the same in
terms of changes made to the HashSet parameter passed to each method.

Each method, benchmark and benchmarkShift achieve the same goal, but go about
it in different ways. Therefore when they are used and placed into their respective
HashSet, the same grams are added.

(2) What are the conceptual differences between the two
benchmarking methods

The main conceptual differences are the focus. In benchmark, the focus was to just complete
the task in the most simple and straightforward method. However, the focus of benchmarkShift
was to keep runtime low and make smart decisiones that would be more efficient even if

it is not as simple.

(3) Is the total amount of memory allocated for arrays the same or different in the two methods? Account for arrays created in the methods and arrays created by WordGram objects. Try to be quantitative in answering.

The amount of memory allocated is more in the benchmark method because the array words is the same size as the number of Strings in the txt file. However in the benchmarkShift, the array words is the size of WSIZE which was at most 10 from the tests I ran. This meant that the amount of effort was drastically more in the benchmark method. In the objects WordGram, the methods used required working with the array words which means that the operations done using benchmark, such as the for each loop in making the String of all the words in the array myWords, was less-efficient.