

Angikar Ghosal ag520

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.

Both benchmark and benchmarkshift achieve the same result, but benchmark uses an ArrayList and benchmarkShift uses an array. 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.

The results of the two methods is the same as the overall outcome is the same. benchmark uses an ArrayList to store and add new words, converts to an array and creates the Wordgram. benchmarkShift directly uses ShiftAdd to add and create from the array. The net elements in the array are same.

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

The benchmark method adds strings to an arraylist and then converts the ArrayList to an array. When it calls the shiftAdd unction, the start values are different from beginning. However, benchmarkShift directly adds strings to the array, and uses shiftAdd directly. The two methods also access memory each other.

(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 total amount of memory allocated for arrays is different. In an ArrayList, when initial memory space is used up, number of elements that can be stored automatically doubles. Therefore, benchmark is less efficient and uses more memory, because benchmarkShift uses only that much memory as needed.