tasks

A0 Neighborhood Score

Letters are scored according to the rarity with which they appear in a given corpus (a large body of texts). Rare letters are worth more than common letters. Specifically, if a given letter makes up more than 10% of the letters in the entire corpus of text, its score is **0**. If the letter makes up [8,10)% of the corpus (at least 8 but not more than 10 percent of the corpus), its score is **1**. If it makes up [6,8)%, its score is **2**. If it makes up [4,6)%, its score is **4**. If it makes up [2,4)%, its score is **8**. If the letter makes up [1,2)%, its score is **16**. Otherwise, the score is **32**. Whitespaces, numbers, and punctuation do not receive scores. Scoring is case-insensitive.

A word's score is the sum of the scores of all its letters.

A word's *k*-neighborhood is the bag of *k* words appearing before it and *k* words appearing after it in the corpus. For instance given the sentence:

```
Colorless green ideas sleep furiously.
```

the words have the following 2-neighborhoods:

```
colorless ↦ [green, ideas]
green ↦ [colorless, ideas, sleep]
ideas ↦ [colorless, green, sleep, furiously]
sleep ↦ [green, ideas, furiously]
furiously ↦ [ideas, sleep]
```

If a word appears in the corpus multiple times, it will have multiple *k*-neighborhoods.

The score of a *k*-neighborhood is the sum of the scores of the words that are members of the *k* neighborhood.

Subtask A0.1 Warm-up

Write a neighborhood scoring program that for each word *w* in a given corpus returns the mean score of *w*'s *k*-neighborhoods. The parameter *k* is configurable.

The scorer outputs a **CSV** text file with two columns. The first column contains all the words in the corpus in alphabetical order. The second column contains the mean *k*-neighborhood scored for each of the words in the first column.

Subtask A0.2 Parallelization

Parallelize the neighborhood scoring program using Java threads.

Subtask A0.3 Benchmark

Compare the execution time of the sequential scorer and the parallel scorer. Use top 100 books of Project Gutenberg as input (https://www.gutenberg.org/browse/scores/top (https://www.gutenberg.org/browse/scores/top)).

Submitting

Write a one-page report summarising the implementation and the