

HOMEWORK

October 1, 2016

Homework 3B: Due on Monday Oct 10, 2016

- This is a replacement of PART B of the original homework 3.

REPLACEMENT PART B: PROGRAMMING ASSIGNMENT (70 Points)

All needed files are found in `hw3.Replacement.zip` from Piazza/Resources. They must be included in your `src` folder but unmodified. Our Makefile should also work for `run1` and `run2`.

Question B.1 (30 Points) NewArrayStringLog and NewZoombinis

First, read the original hw3 question about ArrayStringLog, and study our (accidentally) provided implementation, found in the file `src/stringLog/MyArrayStringLog.java`. Also read the original hw3 question about Zoombinis names, and study our (accidentally) provided implementation, found in the file `src/zoombinis/Zoombinis.java`. You can call both programs with our targets `run1` and `run2`. Keep our implementation files unchanged.

We now want you to modify the `MyArrayStringLog` class so that, instead of inserting random numbers, you insert random Zoombini names. However, our original `Zoombinis` class must be modified to support this. Therefore, you will need to write modified versions of these files in `src/stringLog/NewArrayStringLog.java` and `src/zoombinis/NewZoombinis.java`, respectively.

- (a) `NewArrayStringLog` has a main method whose command line arguments are `s,n,m` as before, but adds a fourth command line argument called `mode`. The default values for these arguments are

`s=0, n=600, m=5, mode=1`

The actions of the main method are the same as our original implementation, except that you insert random `NewZoombini` names of the specified `mode` into our `NewArrayStringLog` instance.

- (b) In addition to `run1` and `run2` which we have provided in the Makefile, you must now add the target called `run3`. This target is analogous to `run1`, but it runs the main method of `NewArrayStringLog`.

Question B.2 (10 Points) Zoombinis!!!

Please create new syntax file in any mode of your choice (except modes 0, 1, 2, 7 which we have already provided). Create a new target `run4` in your Makefile (analogous to `run2`) to show off your new mode. Please call the main program in our original `Zoombinis` class (not your `NewZoombinis` class for previous question).

Be sure to explain in the `info` file of your syntax folder your rationale for this design.

Here is a thought about your goal in generating names: you are trying to strike a compromise between **specificity** and **coverage**. Specificity means that you want to ensure that only valid names of that culture are generated. Coverage means that you want to ensure that no valid names are omitted.

E.g., you have very good specificity in Chinese names if your LDT only allows 2 syllable names, only allows names with initial consonants (but limited to **Ch**, **Zh**, **Sh**), and with terminal consonants (but limited to **n**, **ng**). But you would have bad coverage (you will miss common Chinese names ending with the syllable **li**).

Question B.3 (20 Points) Linked Lists

We have provided a class in `src/linkedList/Node2Abstract.java` with a main method which you must not modify. This is a generic class (parametrized by type `T`) for maintaining **doubly-linked lists** (the "2" in `Node2` suggests the double). Details for this class are found in the file. Please write a class in `src/linkedList/Node2` to extend our abstract class, but do not write any main method. In your Makefile, please write a target called `run4` (emulating `run1`) to call the main method of `Node2` (which is just inherited from `Node2Abstract`).