Data Structures, IDC Herzliya, 2017

Assignment 2: The German tank problem

During the Second World War, the allied forces became very interested in the production capabilities of German tanks. Specifically, they wished to answer the question, how many Panzer tanks could the German industry produce in a single month? Standard intelligence gathering techniques proved inefficient and suggested a production rate which was much higher than the real German capabilities. Luckily, the allied forces knew statistics.

British statisticians inspected German tanks which were captured in battle. They've noticed that each tank carried a serial number. Since the serial numbers were ordered in a consecutive manner, knowing the highest serial number would reveal exactly the amount of produced tanks. Of course, the allied forces could never know for sure that they've managed to capture the tank with the highest serial number. Instead, the highest serial number was given an estimation, based on the population of captured tanks.

The estimation followed along these lines. Suppose k German tanks were captured, and denote by m the maximal serial number observed among captured tanks. Then an estimation for the total number of produced tanks is given by:

$$m + \frac{m}{k} - 1.$$

As simple as this formula may appear, it consistently gave estimation with an error rate of less than 10%. In comparison, standard intelligence estimated that the German could produce more than 5 times the actual number of tanks.

In this assignment you will simulate the work of the British statisticians. You will implement several classes for the simulation. The classes are supplied as skeletons and you need to fill in the functions. Note that some functions require will require you to implement them with a certain bound on the runtime complexity.

You are not allowed to add any public functions but may add any number of private functions. **Do not use any external libraries or import unnesscary packages.**

We will now give a short description of all classes.

Tank

The tank class represents a German tank. Tanks are characterized by a String, representing their serial number. Such strings are composed of letters in the English alphabet, which may or may not be capitalized. Each letter in the string represents a number between 0 and 25 , with a corresponding to 0 and z corresponding to 25 . Thus, the entire string is a positive integer which is written in base 26 . Note that the description given above is case insensitive. So, the same number can be represented by several different strings. Below are some examples of possible strings and the integers they represent.

```
"A" is the number 0


"b" is the number 1
```

"ba" is the number 26. To see this we multiply the 'values' of b

and a

by appropriate powers of 26. $1 \cdot 26^1 + 0 \cdot 26^0 = 26$

"Ba" is also 26

"dBb" is 2055, since $3 \cdot 26^2 + 1 \cdot 26^1 + 1 \cdot 26^0 = 2055$

Heap

Since the estimation of the total number of produced tanks relies heavily on knowing the maximal serial number among all serial numbers of captured tanks, we will use a max-heap data structure to store information about captured tanks. The implementation of the heap should use an internal array of Tanks. The tanks are ordered in the heap according to their serial numbers. That is, for each node in the heap, the serial number of the tank in the given node must be bigger than the serial number of the tanks stored in the descendants of the node. The main purpose of the heap is to store the information of an increasing number of captured tanks and allow at each moment in time to know the maximal serial number observed. You may assume the maximal serial number will never be bigger than $10,000$.

Remark: You are asked to implement a general heap data structure. Some of the functions you will write are not directly applicable to the problem at hand.

TankEstimator

The estimator class wraps the max-heap and adds the functionality of estimating the number of German tanks produced. Estimating this number is done according to the formula given in the introduction.

Simulation

The main function in this class takes as input three integers from the command line, N ,k and T . N is the total number of Tanks produced by the German industry, k is the total number of tanks captured by the allied forces and T is the total number of experiment to be run. At each experiment the simulation first creates an instance of *TankEstimator*, then it chooses at random k **different** serial numbers, from all possible serial numbers in the range 1 to N , and stores tanks with the appropriate serial number in the *TankEstimator* instance. At the end, it calculates an estimation to N . After running T experiments, the program prints the average estimation obtained in all experiments.

Experiment with different values of N , k and T to understand when is the given estimation satisfactory. For example, try $N = 1000, k = 200, T = 1000$.

Submission

Before submitting this assignment, take some time to inspect your code, check that your functions are short and precise. If you find some repeated code, consider making it into a (private) function.

Make sure your code is presentable and is written in good format. Any deviations from these guidelines will result in a point penalty.

Submit a zip file with he following files only:

- `Tank.java`
- `Heap.java`
- `TankEstimator.java`
- `Simulation.java`

The name of the zip file must be in the following format "ID.zip", where "ID" is your id. For example, "03545116.zip".

Deadline: Submit your assignment no later than May 12, 23:55.