Programming Assignment 1

CSCD320 Algorithms

Instructor: Dr. Bojian Xu
Eastern Washington University, Cheney, Washington

Due: 11:59pm, May 11, 2016 (Wednesday)

Please follow these rules strictly:

1. Verbal discussions with classmates are encouraged, but each student must independently write his/her own work, without referring to anybody else's solution.
2. No one should give his/her code to anyone else.
3. The deadline is sharp. Late submissions will **NOT** be accepted (it is set on the Canvas system). Send in whatever you have by the deadline.
4. Every source code file must have the author's name on the top.
5. All source code must be written in Java and commented reasonably well.
6. You are not allowed to use library-provided subroutines if they are what you are asked to implement.
7. Sharing any content of this assignment and its keys in any way with anyone who is not in this class of this quarter is NOT permitted.

Finding the Richest People

IRS wants to find the top $10,000$ richest people and double checks their tax return reports. What IRS has is a file that contains the list of all incomes along with the tax payers' information, such as SSN, names, tax rate, etc. Unfortunately, the tight budget constraint does not allow IRS to maintain large-memory supercomputers. Instead, what IRS has is a normal computer whose main memory capacity is just large enough to load the information of $10,000$ people, but the length of the list in that file is way longer than $10,000$.

You are an expert in the IT department of IRS and are consulted for an efficient solution to find the top $10,000$ richest people from that list using that cheap computer. After reading the challenge, as a computer science graduate, you become to understand that the above challenge can actually be modeled as the following computer science algorithm problem.

Specification:

- The Java class that has the *main* function needs to be named as "Richest". That is, the name of the source code file that contains the *main* function should be "Richest.java".

- The input to your algorithm is a text file, where each line is an integer number. The name of the input file will be provided by the grader as a command line parameter. For example, if the grader provides an input file named "`irs.txt`", the command line to run your program will be: `$java Richest irs.txt`

  (Note: `$` is the command line prompt and is not part of the command line.)

- Your program's output should be the top $10,000$ largest numbers in that file. Save the output in a file named "richest.output" in the working home directory (the directory where your source code is located). Make sure each line of the file contains only one number and all numbers are sorted in **descending** order in the end.

- You program cannot load more than $10,000$ numbers in the main memory at any time of the run of your program. Of course, you are allowed to use some other constant (and negligible) amount of space in your program for some variables.

Design and implement a time- and space-efficient algorithm to solve this algorithmic challenge.

Hint: First, use a min-heap, which physically is a 1-d integer array, to find the top $10,000$ largest number. After those largest numbers are found, use the heap sort to the sort the physical array in descending order and then flush the content out to the disk file.

Note: You must write every line of the code by yourself, including those operations of the heap data structure (heapify, heap sort, etc.). You CANNOT use any library provided heap data structures and/or subroutines. Otherwise, it will not make any sense to do this assignment and you will lose a huge chunk of points.

For your convenience, I attached `datagen.c`, a simple `C` program that you can use to generate a data file for you. You feel free to change the parameter in the code to control the size of the file that you want to generate. You even don't have to use it.

Submission

- All your work files must be saved in one folder, named: **firstname_lastname_EWUID_cscd320_prog1**
  (1) We use the underline '_' not the dash '-'.
  (2) All letters are in the lower case including your name's initial letters.
  (3) If you have middle name(s), you don't have to put them into the submission's filename.
  (4) If your name contains the dash symbol '-', you can keep them.
  (5) You do NOT need enclose the testing data files and the Java .class files.

- You then compress the above whole folder into a .zip file.

- Submit .zip file onto the Canvas system by the deadline.