Programming Assignment #2

Programming assignments are to be done individually. You may discuss the problem and general concepts with other students, but there should be no sharing of code. You may not submit code other than that which you write yourself or is provided with the assignment. This restriction specifically prohibits downloading code from the Internet. If any code you submit is in violation of this policy, you will receive no credit for the entire assignment.

Problem Description

In this problem, your job is to find a food airdrop program for the lions in a national park. The lions in the park cannot find enough food due to a shortage, so we are planning to feed them with airdrops. The lions live in communities so each community should be supplied with one airdrop.

For this problem, you are given the map of lions in the national park as a 2D array. Each cell of the array corresponds to a region of the park and value of the cell corresponds to number of lions living in this region. A region may be connected to another region vertically and horizontally, not diagonally. Also, for the connection, both regions should have at least 1 lion living.

For each set of connected regions, an airdrop with size equal to the number of total lions in the region should be sent. The location of the airdrop should be the region in the set with smallest $row + column$ value. If there are 2 smallest regions, such as $(1, 2)$ and $(2, 1)$, the one with smaller row number $(1, 2)$ should be chosen. Consider the following example:

$$Map = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 2 & 1 \\ 3 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 \end{bmatrix}$$

In this example, number of lions living in $(0, 0)$ region is 1 and number of lions living in $(2, 1)$ region is 0. Total number of lions is 14, and there are 4 connected regions where the populations are $\{(1, 1), (3, 2), (1, 2, 1), (3)\}$. The result will be:

- Airdrop at $Region(0, 0)$ with size 2
- Airdrop at $Region(2, 0)$ with size 5
- Airdrop at $Region(0, 3)$ with size 4
- Airdrop at $Region(3, 2)$ with size 3

The result should be sorted with respect to $row + column$ value. Again, if there is a tie, the one with smaller row value should be chosen.

The complexity of your solution should be linear with size of the map. If number of rows $= n$ and number of columns $= m$, the complexity should be $O(m \times n)$

Instructions

- For this assignment, you are provided with `Assignment2.java` file, which includes the header of the function which needs to be implemented. You are also provided with `Airdrop.java` class, which has 3 fields as number of rows, number of columns and size of the airdrop. The function will return an array of Airdrops.
- You can only use java.util.* classes and the classes we supplied. Your `.java` file should not import any libray. Since we are not compiling your code with any other dependency, your code will not be compiled if you use any other library.
- Please do not add any package statement to your `.java` file. We will be testing your submissions in the default directory.
- Please do not include any test code or any main method in your `Assignment2.java` file.
- You are also provided with sample grading script, please read the instructions inside `.zip` file provided and make sure that your code passes the sample test. Script is written in bash and it uses Maven to build. You can the script in a linux machine where Maven is installed, by typing "sh grader.sh". You can use LRC machines.
- We will be grading in Java 1.8. **It is YOUR responsibility to ensure that your solution compiles with the standard Java 1.8 configuration (JDK 8).** For most (if not all) students this should not be a problem. If you have doubts, email a TA or post your question on Piazza.
- Please make sure that the header of your `Assignment2.java` file includes your necessary information. Fill the header with your information.
- If you do not know how to install Java or are having trouble choosing and running an IDE, post on Piazza for additional assistance or come speak to a TA.
- There are 2 `.java` files as Assignment2.java and Airdrop.java, but you only need to make modifications to `Assignment2.java`. **Do not modify the other files we have given you, we will be using our own when grading.**
- We will be checking programming style. A penalty of up to 10 points will be given for poor programming practices (e.g. do not name your variables foo1, foo2, int1, and int2).

What To Submit

You should submit a single `Assignment2.java` file to Canvas BEFORE 11:59pm on the day it is due.