Hack 13.0

Computer Science I – Java
Searching & Sorting

Department of Computer Science & Engineering
University of Nebraska–Lincoln

Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. Collaboration is allowed and, in fact, *highly encouraged*. You may start on the activity before your hack session, but during the hack session you must either be actively working on this activity or *helping others* work on the activity. You are graded using the same rubric as assignments so documentation, style, design and correctness are all important.

Problem Statement

You will use the implementation of your `Airport` class from a previous hack to develop several reports that will require you to sort and search (a subset of) the International Civil Aviation Organization database for particular airports.

We have provided a source file, `AirportUtils.java` with several comparators and methods you need to implement. Details are provided in the documentation.

The `generateReports()` takes a list of `Airport` s and should produce the following reports which should all be output to the standard output.

- To help you troubleshoot, you should print out the list of airports in the original order to the standard output.
- Sort the airports by each of the 8 comparators and print them out (8 reports total).
- Search for and print out the airport in the array that is closest (via air distance) to Lincoln. Lincoln is located at 40.8507N, 96.7581W.

- Search for and print out the airport that is the geographic west-east median[1] of the given airports with respect to longitude.
- Search for an airport located in the city New York (city would be `New York` and the country would be `US`) and print it out if it exists. If no such airport exists, print out an appropriate message.
- Search for an airport whose type is `large_airport` and print it out if it exists. If no such airport exists, print out an appropriate message.

Instructions

- Hand in both your `Airport.java` and `AirportUtils.java` source files. You may add any utility methods you wish but you must *not* change any of the signatures of the required methods.
- In addition, you must create a `main` method in your `AirportUtils.java` class that tests your reports with at least least 5 airports.
- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.
- You may (in fact are encouraged) to define any additional "helper" methods that may help you.
- Include the name(s) of everyone who worked together on this activity in your source file's header.
- Turn in all of your files via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.

[1] With 0-indexed lists, the median is usually the element at index `n/2` when sorted. This relies on truncation to give the middle index of odd-sized lists and prefers the "right" element for even-sized lists.