

**Due Oct. 19th, 11.55pm**

This project will use the BRIDGES queue API to explore Earthquake data that will be retrieved from a source via Twitter. Instead of the interface you used from the textbook, you will use an equivalent interface from BRIDGES for queue operations.

The BRIDGES API is accessible from <http://bridgesuncc.github.io/doc/api/0.99.0/>

**Dataset:**

We will be using tweets of earthquakes as part of this project. The actual tweets will be accessed by BRIDGES from <https://twitter.com/earthquake>. Each tweet contains the quake magnitude, location, date and time. Utilities will be provided to access some of these attributes for use in the tasks described below.

**Project Tasks:**

You will need to do the following in this project:

1. **Queue Tutorial:** You will be provided with a simple driver that retrieves a few earthquake data records provided by above site, and insert that into the queue. It will also illustrate how to change the properties of queue elements. The queue will then be displayed by BRIDGES on the specified website. You will carefully review the tutorial, including all BRIDGES related calls, for use in the following tasks two described below.

**Task1: You will show the visual output of the two simple drivers on Moodle.**

2. **Earthquake Snapshots.** In this part of the project, you will assume that the queue has a **fixed number of elements**. The goal is to make repeated calls to the data to get more recent records. As new records are enqueued, in order to accommodate the new entries, we might end up having to dequeue older records(records come in chronological order).

**Task2: Assume a maximum queue size of 20. You will initially obtain 20 data items (see the provided driver for guidance for data acquisition calls). After this, request 5 data items each time; you will make 4 such requests. You will update the queue after each request, taking care to ensure the queue does not exceed its maximum size. Use the Bridge.update() function to create a new visualization, each time the queue is updated.**

3. **Filter the Quake Data.** In this part, you will write a new driver to acquire a fixed number of data items. Your goal is to filter the data prior to enqueueing the data. You will filter the data by quake magnitude as well as adjusting the size of the queue items to indicate its magnitude.

**Implementation Details.** Implement this part of the project by e a separate driver or function. The earthquake magnitudes range from 5-9(Richter scale). Assume a maximum size of 50 for the queue. Bridges item sizes range from 0-50 (for practical reasons, use (5-50)). Your job is to map a range of earthquake magnitudes into a size range, so the visualization gets a good range of sizes that correspond to earthquake magnitudes. Run your program for ranges 5-6, 6-7, 7-8. Save each output separately. The provided example driver illustrates how to change the size of a queue item and query for earthquake magnitude.

**Saving Output.**

You can save your output by selecting unique assignment ids, as part of the Bridge.init() call. Each update() call will save the output within this assignment id. If assignment id is 7, then 7.0, 7.01, 7.02... will be part of the links generated to save the visualization(needed in part 2).

**Evaluation:**

An interactive demonstration of the project will be required. Given the server issues and limits of Twitter requests, its best to save the output in web pages prior to the demo. **Note that if you send output to the same page, your old output will be overwritten.** Provide the links to the output webpages in your submission to Moodle, in addition to your source code.