Gary
APCS2 PD8

KDubs EDubs
KimJoshUn

# Project Proposal

---

**Name of Project: Twitter Tree/Heap**

Essentially, we will be using heaps (min and max) and trees to sort tweets of a certain search query in a number of different ways - the user can choose.
The ways to sort tweets, as of now, are as follows:

- By follower count
- By number of retweets
- By proximity to current location
- More may be added, depending on time

Because queries are rate limited (we can only access 180 tweets every 15 minutes), we will also be using a cache of tweets - we will constantly be retrieving tweets and its data, accumulating it locally for later use. This cached data will be heapified, both with minHeap and maxHeap, depending on the selected sort (i.e. if the user elects to sort by maxFollowers, a visual heap will be constructed (in real time) with the cached data, in which the tweet of the user with the most followers will be the root, and subsequent tweets will follow maxHeap specification.)

We will also create a tree of tweets updated in real time, where the user can choose to “get new tweet,” and the most recent tweet will be added to the tree and placed where it should be, based on the sort method selected.

User can also add tweets to a ‘queue,’ and they will be sent out as such.

Using processing, we will visualize the creation of a tree or heap in real-time. Tweets will “swap” with each other when a new tweet is added to the data structure, for example.

Twitter data will be extracted by using a pre existing Java library for the Twitter API:

- http://twitter4j.org/en/

**Structure:**

- Node based tree (or should we use an arraylist)
    - Each node will contain an object that represents a tweet

**Classes/Methods:**

**Tweet** - contains the relevant information for each tweet - *implements Status*
Instance vars:
User author- the user that tweeted the tweet
- Can use User methods to get data on the user itself

geoLocation - the coordinates of where the tweet was sent out

**Tree** - sorted tree that contains specific tweets based on recency and popularity (typecast for different sorts)
    - rearrange() - allows user to rearrange the tweets within the tree (maybe idk about this tho)
    - remove() - allows user to remove a certain tweet and replace it with the next relevant tweet
        - Similar to how elements shift up in binary trees

**Prioritized To-do list:**
- Create main classes
- Create data structure to contain tweets
- Implement twitter4j library to work in conjunction with interface
- Complete basic GUI
    - Have the circles have them expand when clicked on, etc.
- Test each method*

**Timeline**
V1:
- Implement main classes and methods
- have basic GUI done (without data)

V2:
- Integrate twitter data into data structure

V3:
- Display data structure in GUI

V4:
- Add additional methods