## CS 293 Lab Assignment 8 Due date: Oct 7, 2022

In Lab Assignment 6, we had implemented a system wherein users can enter reviews for journeys between a given pair of stations. In that assignment, we allowed the user to search for these reviews using specific keywords.

In this lab, we'll try to implement another feature that users may find useful. Over time, a large number of reviews may accumulate for journey(s) between a given pair of stations, and it may not be possible for a user to guess keywords present in reviews entered by other users. So, we'll now add a numeric rating to each review (on a scale of 0 to 100), and we'd like the user to be able to see a subset of these reviews whose rating is at least as large as a user-provided threshold. However, we'd like to do this without sorting the entire set of reviews (in a real system, reviews may keep coming in even as a user is trying to search for reviews), and without even accessing the entire set of reviews most of the time. The use of priority queues implemented using heaps is the key to getting this done.

As an example, suppose there are 1000 reviews of journeys from MUMBAI CENTRAL to CHENNAI CENTRAL (possibly by various trains, i.e. journey codes) with reviews rated from 0 to 100. A user may now want to see *a sampling of reviews (not necessarily all reviews)* with rating >= 50 of these journeys. The way we will operationalize this is as follows. We will fetch at most 10 reviews (in no fixed order of ratings) of journeys from MUMBAI CENTRAL to CHENNAI CENTRAL at a time, and build a priority queue with the fetched reviews. The priority queue will be implemented as a max-heap, i.e. rating of review at a node is always at least as large as ratings of reviews at its children. If the root of the resulting heap has a review with rating >= 50, then we will show this review to the user, delete this review from the heap (while maintaining the remaining reviews as a heap), and check if the root of the resulting heap still has a rating  $\geq 50$ . If so, we show this review to the user, delete it from the heap, and continue in this manner. If at any time, the rating of a review at the root of the heap becomes < 50, we know that all reviews in the heap now have ratings < 50. At this point, we stop and ask the user if she/he/they want to see more reviews. If the user answers yes, we then fetch the next 10 reviews (in no fixed order of ratings) of journey from MUMBAI CENTRAL to CHENNAI CENTRAL and repeat the above process, building on the heap that we had already built.