Assignment 4
Design and Analysis of Algorithms

Nilanjan Datta

Submission Deadline: 27/03/2022

1. Given an airport with a single runway, design an efficient runway reservation system of that airport. Each reservation request comes with requested landing time let's say t. Landing can go through if there is no landing scheduled within k minutes of requested time, that means t can be added to the set of scheduling landings. k can vary and depends on external conditions. This system helps with reservations for the future landings. Once the plane lands safely, you have to remove the plane for landing sets.

   (a) Suggest a data structure to implement the above system so that new reservations can be done in $O(\log n)$ times, where n is the cardinality of the current set of scheduling landings.

   (b) How can you modify your data structure to report the number of planes scheduled between t_1 to t_2 in $O(\log n)$ time?

2. Suppose IAI wish to build a messaging app **chat@IAI** for all the students, faculties, and administrative staffs, where admins can create different chat rooms. Each user is identified by a known unique integer ID. The chat consists of a linear stream of messages, each written by an user. Everyone can see the most recent k chat messages, where k depends on the size of their screen. If some user misbehaves in chat, he/she gets banned by the admin. When a participant gets banned, he/she can not post any new messages in chat, and all of his/her previously sent messages are removed from the chat. Describe necessary data structures (multiple, if required) to efficiently implement **chat@IAI**, supporting the following operations (all operations should be worst-case), where n is the number of all users (banned or not).

   (i) BUILD(U): Initialize a chat room with the n = $|U|$ users in $O(n \log n)$ time.

   (ii) SEND(v, m): Send message m to the chat from user v (unless banned) in $O(\log n)$ time.

   (iii) RECENT(k): Return the k most recent not-deleted messages (or all if $< k$) in $O(k)$ time.

   (iv) BAN(v): Ban user v and delete all their messages in $O(n_v + \log n)$ time, where n_v is the number of messages that viewer v sent before being banned.