Assignment 5
Design and Analysis of Algorithms

Nilanjan Datta

Submission Deadline: 10/05/2022

1. Design a data structure that maintains a dynamic set S of n elements subject to the following operations and time bounds:

   - INSERT(x, S): Insert x into S in $O(\log n)$ (expected amortized).
   - DELETE(x, S): Delete x from S in $O(\log n)$ (expected amortized).
   - SUCCESSOR(x, S): Find the smallest element in S larger than x in $O(\log n)$ (worst case).
   - FIND-MIN(S): Return the smallest element in S in constant time (worst case).
   - SEARCH(x, S): Return TRUE if element x is in S in constant time (expected).

2. IAI wants to assign $2n$ new students to n cubicles, numbered 0 to $n-1$, in student's workplace. Each student will have an ID: a postive integer less than u, with $u \gg 2n$. No two students can have the same ID, but new students are allowed to choose their own IDs after the start of term. IAI wants to find students quickly given their IDs, so will assign students to rooms by hashing their IDs to a room number. So as not to appear biased, IAI will publish a family $\mathcal{H}$ of hash functions online before the start of term (before new students choose their IDs), and then after students choose IDs, IAI will choose a rooming hash function uniformly at random from $\mathcal{H}$. Two new IAI students S_1 and S_2 want share a cubicle. For each hash family below, show that either they can choose IDs k_1 and k_2 so as to guarantee that they'll be roommates, or prove that no such choice is possible and compute the highest probability they could possibly achieve of being placed in the same cubicle:

   - $\mathcal{H} = \{h_{ab}(k) = (ak + b) \mod n \mid a, b \in \{0, \ldots, n-1\},\ \ a \neq 0\}$.
   - $\mathcal{H} = \{h_a(k) = (\lfloor kn/u \rfloor + a) \mod n \mid a \in \{0, \ldots, u-1\}\}$.
   - $\mathcal{H} = \{h_{ab}(k) = ((ak + b) \mod p) \mod n) | a, b \in \{0, \ldots, p-1\},\ \ a \neq 0\}$, for a fixed prime $p > u$.