Technische Universität München
Fakultät für Informatik
Lehrstuhl für Effiziente Algorithmen
Prof. Dr. Harald Räcke
Chintan Shah

Wintersemester 2015-16
Assignment 8
December 7, 2015

Efficient Algorithms and Datastructures I

Question 1 (10 Points)

Design a data structure that handles the database of a single bank account. This data structure should support insertion of past and future transactions, as well as deletion of existing transactions. Assume that no 2 transactions occur at the same date. You should support the following functions:

(a) INIT() : Initializes the account. The initial balance in the account is 0 €.

(b) INS-TRANS(sum,date) : Insert a given transaction at a given date. The sum can be either positive or negative, and should be added to the balance in the account starting from the following day. Note that the date can be arbitrary (not necessarily today's date).

(c) DEL-TRANS(date) : Delete the transaction that occurs at the given date, if there is any. When a transaction is deleted, the corresponding sum should be subtracted from the balance in the account starting from the following day.

(d) BALANCE(date) : Returns the balance in the account at the beginning of the given date.

You should be able to perform INIT() in $O(1)$ time and all other operations in $O(\log n)$ time, where n is the number of transactions.

Question 2 (10 Points)

Suggest how to use a skip list so that given a pointer to a node with key x, we can return a pointer to a node with key $y < x$ in $O(\log k)$ expected time where k is the distance between the nodes with values y and x in L_0.

Question 3 (10 Points)

In hashing with chaining, if we modify the chaining scheme so that each list is kept in sorted order, how does it affect the running time for successful searches, unsuccessful searches, insertions, and deletions?

Question 4 (10 Points)

(Extra Question)

A forest is an undirected cycle-free graph, i.e. a forest is a graph, all of whose connected components are trees. A random graph is obtained by starting with a set of n vertices and adding edges between them at random. In the $G(n, p)$ model, for every pair of vertices $\{a, b\}$, the edge (a, b) occurs independently with probability p. Your goal is to show that for suitable value of p, the probability that G is not a forest is at most a constant (say $\leq \frac{1}{2}$). Note that G not being a forest means that $\exists$ a set $S \subseteq V, |S| = k$, such that the graph induced by S contains at least k edges. Of course, this trivially holds if for example you choose $p = 0$. You should aim for $p = \Omega\left(\frac{1}{n}\right)$. For example, $p = \frac{1}{4e^2 n}$ might be a good choice.