Technische Universität München
Fakultät für Informatik
Lehrstuhl für Theoretische Informatik
Prof. Dr. Harald Räcke
Richard Stotz, Dennis Kraft

Winter term 2016 / 2017
Assignment 7
December 5, 2016

Efficient Algorithms and Data Structures I

Deadline: December 12, 2016, 10:15 am in the ***Efficient Algorithms*** *mailbox.*

Homework 1 (5 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. Prove that your method works!

Homework 2 (6 Points)

Amanda is developing the app `PrintNeighbors`, that allows users to share their printer with others for a small fee. She decides to use her favorite data structure, Skip Lists, to manage the users. Her boss, however, doesn't like randomized algorithms and asks her to *derandomize* the skip list.
Amanda first tries to design a 1-Skiplist, defined as follows. List L_0 contains all elements. For $k \geq 1$, every second element in L_{k-1} is skipped in L_k, thus dividing the number of elements by 2.

1. Analyze the running time of a search in a 1-Skiplist. Show that inserts cannot be done efficiently in a 1-Skiplist.
2. Amanda wants to improve her data structure to support efficient inserts. Describe a deterministic data structure based on skip lists, that can perform searches and inserts in $\mathcal{O}(\log n)$.

   Hint: Think about (a, b)-trees! Relax the number of elements skipped for the next level!

Homework 3 (4 Points)

1. A major drawback of hashing with chaining is the waste of space due to empty table entries. Assume that the hashing function is chosen at uniformly at random from the set of all hash functions (uniform hashing). Determine the expected number of empty table entries in a table of size m after n different inserts. Your result should be a function of m and n.
2. In order to save space, a hash table may store the first element of the chain in the table, instead of storing just a pointer to the chain. Explain why this might be a bad idea using a concrete scenario (fix some reasonably large m and n, as well as pointer and item sizes). In this exercise, you don't need to consider cache-efficiency.

Homework 4 (5 Points)

In double hashing we use the hash function

$$h(k, i) = (h_1(k) + i \cdot h_2(k)) \bmod n \ .$$

Show that when n and $h_2(k)$ have greatest common divisor $d \geq 1$ for some key k, then an unsuccessful search for key k in a full table examines $\frac{1}{d}$th of the hash table before returning to slot $h_1(k)$.

Tutorial Exercise 1

Let $U = \{0, \dots, p-1\}$ for a prime p. For $x \in \mathbb{Z}_p$, define the hash function $h_{a,b}(x)$ as

$$h_{a,b}(x) = (ax + b \bmod p) \bmod n$$

Consider the class of hash functions

$$\mathcal{H} = \{h_{a,b} | a, b \in \mathbb{Z}_p\}$$

(a) Show that $\mathcal{H}$ is not universal.

(b) Show that $\mathcal{H}$ is $(1.1, 2)$ independent for $n \geq 2$ and $p \geq 100n$.

(c) Why would you not choose $\mathcal{H}$ as a class of hash functions?

```
Somehow the verb 'to hash' magically
became standard terminology for key
transformation during the mid-1960s,
yet nobody was rash enough to use
such an undignified word publicly
until 1967.
                          - D. E. Knuth
```