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

Winter term 2016 / 2017
Assignment 8
December 12, 2016

Efficient Algorithms and Data Structures I

Deadline: December 19, 2016, 10:15 am in the ***Efficient Algorithms*** *mailbox.*

Homework 1 (4 Points)

Santa Claus asks you to insert the following values in a Cuckoo-Hashing hashtable. The first hash function is

$$h_1(x) = (3x + 4 \bmod 13) \bmod 9 \;,$$

the second hash function is

$$h_2(x) = (2x + 1 \bmod 17) \bmod 9 \;.$$

Both hash tables use size 9, `maxsteps` is set to 5. Initially, the hash table looks as follows:

	0	1	2	3	4	5	6	7	8
T_1	6		8	7				66	

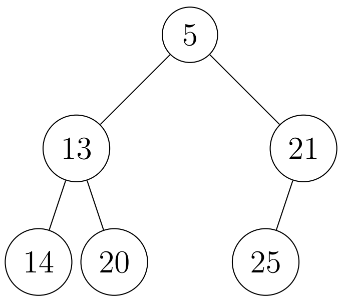
	0	1	2	3	4	5	6	7	8
T_2						19			

Santa wants to see how the hash table looks like after each insert.

1. Insert 10
2. Insert 17
3. Insert 1

Homework 2 (5 Points)

Cueball is preparing christmas by arranging the presents in a binary heap. All presents have an integer written on them. Initially, his heap looks like as follows:



1. The heap can be stored in an array, as explained in the lecture (Slide 319). Show how this array looks like.
2. Some relatives arrive late with their presents and other relatives even retract their present. Cueball has to update his heap! Show how the heap looks like after each operation (as a tree, not as an array).
   (a) Insert 10
   (b) Insert 1
   (c) Delete 5

Homework 3 (5 Points)

During his christmas vacation, App-developer Andy decides to go hiking in the bavarian alps. In a dark forest, he meets a Krampus. The Krampus threatens to punish him for writing another useless app, unless Andy proves his knowledge about Cuckoo Hashing. Andy is asked to prove the following statement:
It is possible to distribute any n keys without collision (i.e. each key is mapped to one of its two valid positions) if and only if there is no set S of keys with $|S| \leq n$ so that the keys in S have at most $|S| - 1$ alternative positions in the two hash tables.
Note that each key has exactly two positions, one for each hash table. You may assume that no two keys have both positions identical.
Example: If key k_1 has position 1 in table T_1 and position 4 in table T_2, while key k_2 has positions 6 and 4 in T_1 and T_2, respectively, then k_1 and k_2 have 3 alternative positions.

Homework 4 (6 Points)

Santa's elves store the children's wish lists in a hash table that has one slot for every child on earth. However, the elf responsible for computing the hash function has had too much Feuerzangenbowle and has forgotten his hash function. Instead, he chooses a position uniformly at random for every wish list, independent of all other choices.
Assume that every child send exactly one list. The number of slots equals the number of lists and is called n. We want to show that the slot with the maximum number of wish lists contains $\mathcal{O}(\frac{\log n}{\log\log n})$ lists.

1. Let X be the maximum height of a single slot (i.e. the number of wish lists in this slot) and let X_i denote the height of the ith slot. How is the random variable X_i distributed? Show that $\Pr[X < t] \geq 1 - 1/n$ is implied by
$$\Pr[X_1 \geq t] \leq \frac{1}{n^2} \ .$$
2. Show that there is a constant $c > 0$ such that for n sufficiently large, we have
$$\Pr\left[X_1 \geq c \cdot \frac{\log n}{\log\log n}\right] \leq \frac{1}{n^2}$$
   Hint: There are several ways to prove this. Chernoff-Bounds or the bounds on Slide 216 are helpful. Use some (constant) base for the logarithm that suits your proof (e.g. base 2 or base e).

Bonus Homework 1 (5 Bonus Points)

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.

Tutorial Exercise 1

Consider a binary heap H implemented with a binary tree data structure (as implemented in the lectures) containing n items. Design an algorithm to find the k-th smallest item in H in $O(k \log k)$ time.

`So what is holding back cuckoo hashing? [...] all that moving around of items.`

`- M. Mitzenmacher`