Technische Universität München
Fakultät für Informatik
Lehrstuhl für Algorithmen und Komplexität
Prof. Dr. Harald Räcke
Richard Stotz

Winter term 2018/19
Assignment 9
December 10, 2018

Efficient Algorithms and Data Structures I

*Deadline: December 17, 2018, 10:15 am in the **Efficient Algorithms** mailbox.*

Homework 1 (5 Points)

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

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

the second hash function is

$$h_2(x) = (2x + 2 \bmod 19) \bmod 9 \ .$$

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		8	17						19

	0	1	2	3	4	5	6	7	8
T_2		4					97		

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

(a) Insert 9

(b) Insert 11

(c) Insert 22

Homework 2 (6 Points)

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 $|S| \le n$ 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 3 (5 Points)

A class $\mathcal{H}$ of hash functions from a finite set U into $\{0, 1, \ldots, n-1\}$ with $|U|, n > 1$ is ε-*universal* if for all $u_1, u_2 \in U$ with $u_1 \neq u_2$

$$\Pr[h(u_1) = h(u_2)] \leq \varepsilon \ ,$$

where the probability is over the choice of the hash function h drawn uniformly at random from the family $\mathcal{H}$.

Show that for any ε-universal family of hash functions, we have

$$\varepsilon \geq \frac{1}{n} - \frac{1}{|U|} \ .$$

Homework 4 (4 Points)

Consider a bipartite (multi-)graph with partitions A, B where $|A| = |B| = n$. Let there be $m = \Theta(n)$ edges in this graph, each edge being chosen uniformly at random (i.e. there may be more than one edge between two vertices). In this graph, find, asymptotically, the expected number of

(a) 2-cycles,

(b) 3-cycles,

(c) 4-cycles.

Note: If we let n be the size of one hash table and m be the number of keys, then the above question asks for the number of 2-cycles, 3-cycles and 4-cycles in cuckoo hashing where each edge in the graph denotes the 2 hash values of a function.

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.

Tutorial Exercise 2

A *soft-heap* is a priority queue that performs insert and delete-min in $\mathcal{O}(\log 1/\varepsilon)$ steps for any $0 < \varepsilon < 1/2$. This is achieved at the expense of "corrupting" keys, i.e. increasing them: At any time, at most εn keys in the heap are corrupted, where n is the total number of elements ever inserted into the heap. After any operation, the soft heap returns a list of newly corrupted keys.

1. We want to select the kth smallest element in a list of n elements. Let $\varepsilon = 1/3$. Use a soft heap to find an element whose rank is between $n/3$ and $2n/3$ in linear time. Apply this procedure recursively to find the kth smallest element in linear time.

2. Using the result from Homework 2 and Part 1, show that an element of rank k in a binary heap can be returned in $\mathcal{O}(k)$ steps.

```
HASH, x.  There is no definition
for this word - nobody knows what
hash is.
                       - A. Bierce
```