University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
P. Bamberger, P. Schneider

Algorithms and Data Structures
Summer Term 2019
Sample Solution Exercise Sheet 4

Exercise 1: Universal Hashing

Consider a hashtable of size $m = 11$ and let $p = 101$. Consider hash functions of the form $h_{a,b}(x) := \left[(ax+b) \bmod p\right] \bmod m$, which form a $\approx$ 1-universal family[1] $\mathbb{H}_{a,b} = \{h_{a,b} \mid a, b \in \{1, ..., p-1\}\}$. Choose one hash function h from the family $\mathbb{H}_{a,b}$. Then find five *different* keys from the set $\mathbb{U} = \{0, \ldots, 99\}$, such that all keys are mapped to the same table entry. Then select a hash function h' from the family $\mathbb{H}_{a,b}$ randomly (or invent appropriate numbers a, b) and remap all keys into the table.

Sample Solution

We choose

$$h := h_{1,1} = \left[(x+1) \mod 101\right] \mod 11.$$

Moreover we choose the following set of keys: 0,11,22,33,44. Then

$$h(0) = h(11) = h(22) = h(33) = h(44) = 1.$$

Consider e.g. $a = 48, b = 18$, i.e. let $h' := h_{48,18}$. Then

$$h'(0) = 7, h'(11) = 8, h'(22) = 9, h'(33) = 10, h'(44) = 9.$$

The point of this exercise was to demonstrate that in case we have some degenerate set of keys that produces much more collisions than would be expected for a random set of keys, we can always rehash those keys with a random hash function from the universal family and *likely* end up with a number of collisions that is closer to the expectation. Note that even though there are hash functions $h' \neq h$ for which this is not true (e.g. $h_{1,2}$ is just as bad as $h_{1,1}$ in terms of collisions), it is unlikely that we pick these when we choose uniformly at random from $\mathbb{H}_{a,b}$.

Exercise 2: Hashing with Open Addressing - Examples

(a) Let $h(s, j) := h_1(s) - 2j \bmod m$ and let $h_1(x) = x + 2 \bmod m$. Insert the keys 51, 13, 21, 30, 23, 72 into the hash table of size $m = 7$ using linear probing for collision resolution (the table should show the final state).

0	1	2	3	4	5	6

(b) Let $h(s, j) := h_1(s) + j \cdot h_2(s) \bmod m$ and let $h_1(x) = x \bmod m$ and $h_2(x) = 1 + (x \bmod (m-1))$. Insert the keys 28, 59, 47, 13, 39, 69, 12 into the hash table of size $m = 11$ using the double hashing probing technique for collision resolution. The hash table below should show the final state.

[1] For $p \gg m$ both prime.

0	1	2	3	4	5	6	7	8	9	10

(c) Repeat part (a) using the *"ordered hashing"* optimization from the lecture.

(d) Repeat part (b) using the *"Robin-Hood hashing"* optimization from the lecture.

Sample Solution

(a)

30	13	21	72	51	23	
0	1	2	3	4	5	6

(b)

	69	13	47	59	39	28	12			
0	1	2	3	4	5	6	7	8	9	10

(c)

30	13	21	72	23	51	
0	1	2	3	4	5	6

(d)

47 $j=1$	12 $j=0$	69 $j=1$	59 $j=1$	28 $j=1$	39 $j=1$	13 $j=1$				
0	1	2	3	4	5	6	7	8	9	10