Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Federal Institute of Technology at Zurich

Department Informatik
Markus Püschel
Peter Widmayer
Thomas Tschager
Tobias Pröger

17th November 2016

Algorithmen & Datenstrukturen Exercise Sheet 9 AS 16

Hand-in: Thursday, 24th November 2016 before the start of the lecture at 10:00 in the entrance area of ML D28. Please staple all sheets together and use this sheet as the cover page. Fill out the first two fields of the form below.

Exercise class (Room & Day):	
Submitted by:	
Corrected by:	
Bonus points:	

Exercise 9.1 *Open Hashing.*

We consider open hashing with a hash table of size p for a prime p.

a) Decide which of the following functions are useful as a hash function (and which are not), and justify your answer.

- $h(k) =$ Digit sum of k
- $h(k) = k(1 + p^3) \bmod p$
- $h(k) = \lfloor p(rk - \lfloor rk \rfloor) \rfloor$, $r \in \mathbb{R}^+ \setminus \mathbb{Q}$

b) Insert the keys $17, 6, 5, 8, 11, 28, 14, 15$ in this order into an initially empty hash table of size 11. Use open addressing with the hash function $h(k) = k \bmod 11$ and resolve the conflicts using

(i) linear probing

(ii) quadratic probing, and

(iii) double hashing with $h'(k) = 1 + (k \bmod 9)$.

For each method, provide the number of collisions. Which one is best for the above situation?

c) Which problem occurs if the key 17 is removed from the hash tables in b), and how can you resolve it? Which problems occur if many keys are removed from a hash table?

Please turn over.

d) In this task we use the hash function $h(k) = k \bmod p$ and resolve collisions using *double hashing*. Let q be the largest prime smaller than p, $h'(k)$ the second hash function and $s(j,k)$ the probing function. The complete hash function in the j-th step is $h(k) - jh'(k) \bmod p$ if $h'(k)$ is given, and $h(k) - s(j,k) \bmod p$ if $s(j,k)$ is given. Decide which of the following choices of $h'(k)$ and $s(j,k)$ are reasonable (and which are not), and justify your answer.

- $h'(k) = \lceil \ln(k+1) \rceil \bmod q$
- $s(j,k) = k^j \bmod p$
- $s(j,k) = ((k \cdot j) \bmod q) + 1$

e) Which advantage does double hashing have when you compare it to quadratic probing?

Exercise 9.2 *Cuckoo hashing.*

Cuckoo hashing is a hashing technique that guarantees constant time *in worst case* for both query and delete operations. The idea is to use two tables T_1 and T_2 of same size and two hash functions h_1 and h_2. A new key x is inserted at position $h_1(x)$ in T_1. In case of a collision, the previously stored key y is displaced to position $h_2(y)$ in T_2. If this leads to another collision, the next key is again inserted at the appropriate position in T_1, and so on.

In some cases, this procedure continues forever, i.e. the same configuration appears after some steps of key displacements due to collisions. We will illustrate such a case in this exercise.

a) Given two tables of size 5 each and two hash functions $h_1(k) = k \bmod 5$ and $h_2(k) = \lfloor k/5 \rfloor \bmod 5$. Insert the keys $27, 2, 32$ in this order into initially empty hash tables.

b) Find another key such that the insertion leads to an infinite sequence of key displacements (in such a case, one would define two new hash functions, allocate two new tables and store the keys in these tables using the new hash functions).

Exercise 9.3 *Implementing a Queue with Stacks.*

An ordinary stack supports the following two operations in time $\Theta(1)$:

- PUSH(x): puts an object x onto the stack.
- POP: outputs and removes the *last* object that was added to the stack.

Describe how a queue can be implemented using two stacks such that the following operations have amortized running time $\Theta(1)$:

- ENQUEUE(x): adds an object x at the end of the queue.
- DEQUEUE: outputs and removes the object *at the front* of the queue.