Albert-Ludwigs-Universität Freiburg
Institut für Informatik
Dr. Alexander Souza
Thomas Janson

WS 2011/12
Sheet 04

Algorithm Theory

Exercise 1 (Universal hashing) [Points: 5]

In the division method for creating hash functions, we map a key k into one of m slots by taking the remainder of k divided by m. That is, the hash function is

$$h(k) = k \mod m.$$

Consider a version of the division method in which $h(k) = k \mod m$, where $m = 2^p - 1$, $p \in \mathbb{N}$, and $k = k_n...k_1k_0$ is a character string interpreted in radix 2^p. The value of k is then $2^{np}k_n + ... + 2^p k_1 + k_0$. Show that if we can derive string x from string y by permuting its characters, then x and y hash to the same value. Give an example of an application in which this property would be undesirable in a hash function.

Two hints:

1. $2^p = 1 + \sum_{i=0}^{p-1} 2^i$.
2. Show that $2^{pi} - 1$ is divisible by $2^p - 1$ for $i \geq 1$.

Exercise 2 (Perfect hashing) [Points: 5]

Let $U = \{0, ..., 28\}$ and $S = \{1, 5, 7, 8, 9, 13, 20, 22, 24, 25\}$.

1. Use the two-level scheme described in the lecture to build a perfect hash function with $k = 5, N = 29, n = |S| = 10$. For $i = 0, ..., n-1$ determine the values W_i, b_i, k_i, and h_{k_i}.

hash value i	keys mapped to hash value $W_i = \{x \in S : h_5(x) = i\}$	#keys b_i	2nd hash table size $m_i = 2b_i(b_i - 1) + 1$	s_i
0				
1				
2				
3				
4				
5				
6				
7				
8				
9				

2. Give the hash table for S.

Exercise 3 (Amortized Analysis) [Points: 5]

Consider a stairway with two allowed operations

1. Walk up one stair with cost 1.
2. Completely go down to the bottom of the stairway which then costs the number of stairs from the current stair to the ground.

Tasks:

1. What is the amortized cost for n operations?
2. Are the amortized cost for n operations still the same when we are also able to step up several stairs in one single operation?

Justify your answers.

Exercise 4 (Amortized Analysis) [Points: 2]

Consider the bit counter example from the lecture. A counter using an array with k bits is counted upwards in an INCREMENT operation starting at 0. Show that if a DECREMENT operation were included in the k-bit counter example, n operations could cost as much as $\Theta(nk)$ time.

Exercise 5 (Amortized Analysis) [Points: 3]

Suppose we perform a sequence of n operations on a data structure in which the ith operation costs i if i is an exact power of 2, and 1 otherwise. Use aggregate analysis to determine the amortized cost per operation.