Technische Universität München
Institut für Informatik
Lehrstuhl für Effiziente Algorithmen
Lehrstuhl für numerische Programmierung

WS 2002/2003
Worksheet 9

3.2.2003

Fundamental Algorithms

Exercise 1

Let $n = 1000$. Compute the values of the hash function $h(k) = \lfloor n(ak - \lfloor ak \rfloor) \rfloor$ for the keys $k \in \{61, 62, 63, 64, 65\}$, using $a = \frac{\sqrt{5}-1}{2}$.

Solution:

$$\begin{aligned} h(61) &= 700 \\ h(62) &= 318 \\ h(63) &= 936 \\ h(64) &= 554 \\ h(65) &= 172 \end{aligned}$$

Exercise 2

Given is a hash table `T[0,...8]` of 10 elements. Draw an image of this hash table after the keys 5, 28, 19, 15, 20, 33, 12, 17, and 10 have been inserted (in that particular order). Use the hash function $h: U \to \{0, 1, \ldots, 8\}$, $h(k) = k \bmod 9$, and use chaining to resolve collisions.

Solution:

`i`	0	1	2	3	4	5	6	7	8
`T[i]`	[]	[10,19,28]	[20]	[12]	[]	[5]	[33,15]	[]	[17]

The []-notation denotes the lists that are stored in each hash table slot.

Exercise 2a

Repeat exercise 2 for hash tables that use open addressing. Use a hash table `T[0,...10]` with 11 elements, instead, and use the following hash functions:

(1) $h(k,i) := (k+i) \bmod 11$

(2) $h(k,i) := \left(k \bmod 11 + 2i + i^2\right) \bmod 11$

(3) $h(k,i) := (k \bmod 11 + i\,(k \bmod 7)) \bmod 11$

Insert the keys 5, 19, 27, 15, 30, 34, 26, 12, and 21 (in that order). State which keys require the longest probe sequence in the resulting tables.

Solution:

(1) linear probing, using $h(k,i) := (k+i) \bmod 11$:

i	0	1	2	3	4	5	6	7	8	9	10
T[i]		34	12		15	5	27	26	19	30	21

Longest probe sequence is 4: for 26

(2) quadratic probing, using $h(k,i) := \left(k \bmod 11 + 2i + i^2\right) \bmod 11$:

i	0	1	2	3	4	5	6	7	8	9	10
T[i]	30	34	27		15	5		26	19	12	21

Longest probe sequence is 2: for 27 and 12

(3) double hashing probing, using $h(k,i) := (k \bmod 11 + i\,(k \bmod 7 + 1)) \bmod 11$

i	0	1	2	3	4	5	6	7	8	9	10
T[i]	30	27	12	26	15	5		34	19		21

Largest probe sequences are 5 (for 34), and 3 (for 12).

Note: Contrary to this example, double hashing usually beats linear or quadratic probing. Moreover, we'd recommend using a larger table for open addressing ...

Exercise 3

Consider a universe U of keys, where $|U| > mn$, and a hash function $h\colon U \to \{0, 1, \ldots, n-1\}$. Show that there is at least one subset of U that contains m keys that are all hashed to the same slot by h.

Solution: proof by contradiction

Assume the opposite, i.e. that for all n values of the hash function the number of elements in U that are hashed to this value is smaller than m. As a consequence, the number of elements that are hashed to any of the n keys is smaller than nm. This contradicts the fact that U is considered to have more than nm elements. Hence, our assumption has to be false, and there has to be at least one subset containing at least m elements.