Hashtable

Hashtables

Problem 1. Consider a hashtable with separate chaining, $M = 10$ slots with integer keys, and hash function $h(x) = x \mod M$.

1. Starting from an *empty* table, show the resulting table after inserting the keys: 3441, 3412, 498, 1983, 4893, 3782, 3722, 3313, 4830, 2001 3202, 365, 12818, 7812, 1299, 999 and 18267.
2. Show the resulting table after applying a rehash of the previous table into 20 slots.

Problem 2. Build the hashtable with separate chaining for the keys 30, 20, 56, 75, 31 and 19, with $M = 11$ slots and hash function $h(x) = x \mod 11$.

Problem 3. The primary drawback of the hashtable technique is that the size of the table to be fixed at a time when the actual number of entries is not known. Assume that your computer system incorporates a dynamic storage allocation mechanism that allows to obtain storage at any time. Hence, when the hashtable H is full (or when the load factor is over certain threshold), a larger table H' is generated, and all keys in H are transferred to H', whereafter the store for H can be returned to the storage administration. This is called rehashing. Write an algorithm that performs a rehash of a table H of size N.

Problem 4.

(a) Insert integer keys $A = \{47, 61, 36, 52, 56, 33, 92\}$ in order into a hashtable of size 7 using the hash function $h(k) = (10k + 4) mod 7$. Each slot of the hashtable stores a linked list of the keys hashing to that slot, with later insertions being appended to the end of the list. Draw a picture of the hashtable after all keys have been inserted.

(b) Suppose the hash function were instead $h(k) = ((10k + 4) \bmod c) \bmod 7$ for some positive integer c. Find the smallest value of c such that no collisions occur when inserting the keys from A.

Problem 5. Farryl Dilbin is a forklift operator at Munder Difflin paper company's central warehouse. She needs to ship exactly r reams of paper to a customer. In the warehouse are n boxes of paper, each one foot in width, lined up side-by-side covering an n-foot wall. Each box contains a known positive integer number of reams, where no two boxes contain the same number of reams. Let $B = \{b_0, \ldots, b_{n-1}\}$ be the number of reams per box, where box i located i feet from the left end of the wall contains b_i reams of paper, where $b_i \neq b_j$ for all $i \neq j$. To minimize her effort, Pharryl wants to know whether there is a close pair (b_i, b_j) of boxes, meaning that $|i - j| < n/10$, that will fulfill order r, meaning that $b_i + b_j = r$.

(a) Given B and r, describe an expected $O(n)$-time algorithm to determine whether B contains a close pair that fulfills order r.

(b) Now suppose that $r < n^2$. Describe a worst-case $O(n)$-time algorithm to determine whether B contains a close pair that fulfills order r.

References

Atserias, Albert et al. (2022). *Data Structure and Algorithms Problem Set*. Universitat Politécnica de Catalunya.

Wirth, Niklaus (1978). *Algorithms + Data Structures = Programs*. (Oberon version: August 2004). Upper Saddle River, NJ, USA: Prentice Hall. ISBN: 0130224189.