DEPT. OF ELECTRONICS AND COMMUNICATIONS ENGINEERING FACULTY OF ENGINEERING CAIRO UNIVERSITY GIZA, EGYPT



Third Year --- Computers: Data Structures May 2011

## Sheet 5: Tables and Hashing

- 1. Consider a hash table of size 13. Draw a representation of a chained hash table if entries with the following keys are added to the table in the sequence provided: 2, 13, 15, 3, 26, 5, 18, 16.
- 2. After insertions are done in problem 1, find how many steps are needed when the following lookup operations are performed on the table: lookup(9), lookup (26), lookup(16).
- 3. In order for hash tables to work with keys that are of type "String", a suitable hashing function must be defined for Strings. Write a function hash(String S, int hashSize) that will provide a reasonable hash value for a string of ASCIII characters. Optional: (advanced C++ programmers can also think about overloading the "%" operator to work on strings.)
- 4. Provide an implementation for hash tables using linear probing for handling the case of collisions (i.e. more than one key mapping to the same home address).
- 5. Rewrite the chained hash table definition and implementation to reuse linked list directly (instead or reimplementing linked lists within the chained hash table). Discuss relative advantages and disadvantages of reusing linked lists versus writing one from scratch within the chained hash table.
- 6. **Programming Experiment**: Consider keys that are integers in the range [1 1,000,000] that will be mapped to a hash table with expected population size of 1000. Generate 1000 random keys in the range of the keys and then generate the hash value using the following two hash functions:
  - a. H1(k) = k % 2000 that maps the keys into a table of maximum size 2000
  - b. H2(k) = k % 2003 that maps the keys into a table of maximum size 2003 (2003 is closest prime to 2000).

Repeat the experiment 10 times using different random seeds. Count the number of collisions resulting from the two hash functions. Comment on the result.

## Some sites related to hashing and prime numbers:

http://primes.utm.edu/ http://www.math.utah.edu/~pa/math/primelist.html http://www.partow.net/programming/hashfunctions/