CEN235 – DATA STRUCTURES
Fall – 2018

Lab 10: Hashing: Linear Probing & Quadratic Probing

In this LAB, you are going to develop two programs for the implementation of both linear probing and quadratic probing.

Preliminary Work:

The process of mapping an object or a number onto an integer in a given range is called hashing.

Problem: multiple objects may hash to the same value. Such an event is termed a collision Hash tables use a hash function together with a mechanism for dealing with collisions.

Hash Function: A function that converts a given big phone number to a small practical integer value. The mapped integer value is used as an index in hash table. In simple terms, a hash function maps a big number or string to a small integer that can be used as index in hash table.
A good hash function should have following properties:
1)Efficiently computable.
2) Should uniformly distribute the keys (Each table position equally likely for each key)
For example, for phone numbers a bad hash function is to take first three digits. A better function is considering last three digits. Please note that this may not be the best hash function. There may be better ways.

Hash Table: An array that stores pointers to records corresponding to a given phone number. An entry in hash table is NIL if no existing phone number has hash function value equal to the index for the entry.
Collision Handling: Since a hash function gets us a small number for a big key, there is possibility that two keys result in same value. The situation where a newly inserted key maps to an already occupied slot in hash table is called collision and must be handled using some collision handling technique. Following are the ways to handle collisions:

- **Chaining:** The idea is to make each cell of hash table point to a linked list of records that have same hash function value. Chaining is simple, but requires additional memory outside the table.
- **Open Addressing:** In open addressing, all elements are stored in the hash table itself. Each table entry contains either a record or NIL. When searching for an element, we one by one examine table slots until the desired element is found or it is clear that the element is not in the table.
-

In open addressing, there are numerous strategies for defining the order in which the bins should be searched:

- Linear probing (Task-1)
- Quadratic probing (Task-2)
- Double hashing

Task-1: Linear Probing [50pts]:

Create fourteen random 3-digit Hexadecimal values and store them in an array with the size of sixteen. As the hash function, you should use *number_to_be_inserted % hash_table_size*. Here numbers are 3-digit Hexadecimal values and *hash_table_size* is 16.

An example number set can be in the following:

19A, 207, 3AD, 488, 5BA, 680, 74C, 826, 946, ACD, B32, C8B, DBE, E9C

Write functions for inserting, removing, searching and printing (i.e. displaying) using **linear probing**.

Calculate the load factor load factor λ and the average number of probes.

Task-2: Quadratic Probing [50pts]:

Create fourteen random 3-digit Hexadecimal values and store them in an array with the size of sixteen. As the hash function, you should use *number_to_be_inserted % hash_table_size*. Here numbers are 3-digit Hexadecimal values and *hash_table_size* is 16.

An example number set can be in the following:

19A, 207, 3AD, 488, 5BA, 680, 74C, 826, 946, ACD, B32, C8B, DBE, E9C

Write functions for inserting, removing, searching and printing (i.e. displaying) using **quadratic probing.**

Calculate the load factor load factor λ and the average number of probes.