**Module 5 Critical Thinking Option #1: Linear Probing**

Youssef Jamal Diallo

Colorado State University Global

CSC506 Design and Analysis of Algorithms

Dr. Dong Nguyen

2/18/2024

Hashing is a cryptographic method that transforms data of any length into fixed-length hashes, called hash values. This is done using a hash function, which is generally a cryptographic algorithm. This transformation allows to obtain hash values much shorter and compact than the original values, which reduces the processing time and the computing power required when searching for information.

The hashing process is based on five characteristics that ensure its security and reliability:

* **Determinism**: the hash function must always generate a fixed hash value for the same input.
* **Illegibility**: hashing transforms the original values into hash values that cannot be read.
* **Uniqueness**: each input value must generate a unique hash value.
* **Variability**: minor changes in the input data should produce completely different hash values.
* **Efficiency**: the hashing process must be fast and efficient.

The hash function is used in various areas, such as database management and security, password management, access authentication, digital signature generation, etc. It also allows us to protect personal data, to analyze or prevent file tampering, and to index data to facilitate searches.

Linear probing is a technique used in hashing to handle collisions, which is when two different keys are assigned the same position in the hash table. When a collision occurs, instead of simply giving up and overwriting the new key, a linear search is performed to find the next empty position in the table.

Here is how linear probing works:

* When a collision occurs, a new position is calculated by adding a fixed offset to the initial index.
* If the new position is occupied, the fixed offset is continuously added until an empty position is found.
* Once an empty position is found, the new key is inserted at that location.

**Performance analysis:**

Linear probing can be efficient when the load factor of the hash table is not too high. However, as the table becomes fuller, collisions become more frequent, which leads to an increase in the number of iterations required to find an empty position. This can significantly slow down hashing operations and lead to poor data distribution.

**Rehashing to overcome the drawbacks of linear probing:**

Rehashing is a technique used to solve the performance problems associated with linear probing. When the hash table reaches a certain load threshold, a new, larger table is created, and all keys are reinserted using a new hash function. This reduces the probability of collisions and improves data distribution, which improves overall hashing performance.

The hash table below has a size of 10 and the following keys have been inserted: 12, 23, 45, 67, 89, 101, 112, 123, 134, and 145.

The hash function used is a simple modulo operation, which means that the hash value for a key is calculated by taking the remainder of the key divided by the size of the hash table.

For example, the hash value for the key 12 is calculated as follows:

12 % 10 = 2

This means that the key 12 is inserted at index 2 in the hash table.

When key 23 is inserted, it is hashed to the same index as key 12. Since the index 2 is already occupied, a linear search is performed to find the next empty position.

The next empty position is found at index 3, so key 23 is inserted at that location.

The process continues for the remaining keys.

A graph of a linear proboscis

Description automatically generated with medium confidence

In summary, linear probing is a simple technique for handling collisions in hashing, but it can lead to performance problems as the table becomes full. Rehashing is used to overcome these drawbacks by reorganizing the data in a new, larger hash table, thus providing better key distribution, and improving overall hashing performance.

**References:**

* Geeksforgeeks. Implementing own Hash Table with Open Addressing Linear Probing. Retrieved from https://www.geeksforgeeks.org/implementing-hash-table-open-addressing-linear-probing-cpp/
* Riker-Guevara, Y. G. (2023, February 28). The linear probing method for hashing. LinkedIn. Retrieved from https://www.linkedin.com/pulse/linear-probing-method-hashing-yamil-riker-guevara
* zyBooks. (2023, November 16). Data Structures Essentials with Python Examples - zyBooks. <https://www.zybooks.com/catalog/data-structures-essentials-python/>