**University of Michigan – Dearborn**

**CIS 200 – Computer Science 2**

**Lab# 8**

Quan Le

[lmmquan@umich.edu](mailto:lmmquan@umich.edu)

November 16, 2024

**Table Content**

Contents

[Question 1 3](#_Toc182688205)

[Source Code 3](#_Toc182688206)

[Description 3](#_Toc182688207)

[Structures 4](#_Toc182688208)

[Screenshots 5](#_Toc182688209)

# Question 1

## Source Code

The source code for this question has been uploaded to Canvas as Lab\_8.cpp.

## Description

* Hashing is a technique used to map data of arbitrary size (keys) to fixed-size values (indices) in a hash table. The main purpose is to provide efficient insertion, deletion, and lookup operations
* **Performance**

|  |  |  |  |
| --- | --- | --- | --- |
| **Aspect** | **Open Chaining** | **Linear Probing** | **Quadratic Probing** |
| Collision Handling | Uses linked lists to store multiple keys at a single index. | Resolves collisions by probing subsequent slots. | Probes slots quadratically away from the original index. |
| Space Efficiency | Requires additional space for lists at each bucket. | Only uses array space (no extra memory needed). | Like linear probing but may leave more empty slots. |
| Lookup Performance | Good when the load factor is low; degrades as chains grow. | Fast when table has low load; performance degrades with clustering. | Less clustering than linear probing, but may skip slots, affecting lookup. |

* **Pros and Cons**
  + **Open Chaining**
    - **Pros:**
      * Flexible and scalable.
      * Performs well under high loads.
    - **Cons:**
      * Requires additional memory for linked lists.
      * Poor cache locality can slow operations.
  + **Linear Probing**
    - **Pros:**
      * Simple and fast for low loads.
      * Excellent cache performance.
    - **Cons:**
    - Suffers from primary clustering.
    - Performance degrades quickly with high load factors.
  + **Quadratic Probing**
    - **Pros:**
      * Avoids primary clustering.
      * Better distribution of keys compared to linear probing.
    - **Cons:**
      * Still suffers from secondary clustering.
      * Not guaranteed to find an empty slot without careful table size selection (e.g., prime size).
* **Application**
  + **Open Chaining:**
    - When load factors may exceed 0.75 or when dynamic resizing is needed.
    - Suitable for applications requiring concurrent access, as individual chains can be locked.
  + **Linear Probing:**
    - Best for static hash tables with low load factors (<0.5).
    - Useful in memory-constrained environments where contiguous memory is a priority.
  + **Quadratic Probing:**
    - Suitable for hash tables with moderately low load factors and where primary clustering is problematic.
    - Useful when resizing is not frequent or feasible.

## Structures

* Define a OpenChainingHashTable class with vector<int> table
* Define a LinearProbingHashTable class with int table[TABLE\_SIZE]
* Define a QuadraticProbingHashTable class with int table[TABLE\_SIZE]

## Screenshots

A screenshot of a computer program

Description automatically generatedTesting case

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generated