Lab Assignment 8

Start Assignment

Due Tuesday by 11:59pm **Points** 22 **Submitting** a file upload
Available Nov 3 at 8am - Nov 11 at 11:59pm 9 days

Peer to Peer Hashing Group Project

Make sure you have read and understood

- ***Unit module 10***
- ***C++ Coding Style Guidelines***

before submitting this assignment. Hand in only one submission.

Lab Assignment Objectives

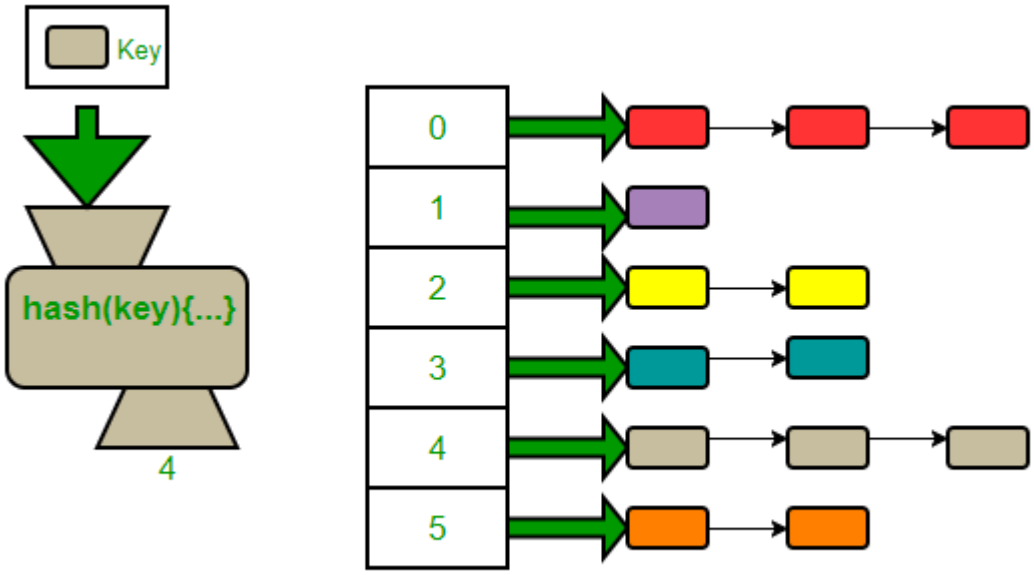
1. Become familiar with MAP and MULTIMAP data structures.
2. Work within a peer to peer group to explore hashing algorithms.
3. Compare and contrast costs and benefits of contrasting chaining and open addressing collision resolution strategies.
4. Analyze performance tradeoffs for bucket sizes used in hashing.
5. Use Big-O notation to describe algorithm run time.

Understand the Assignment

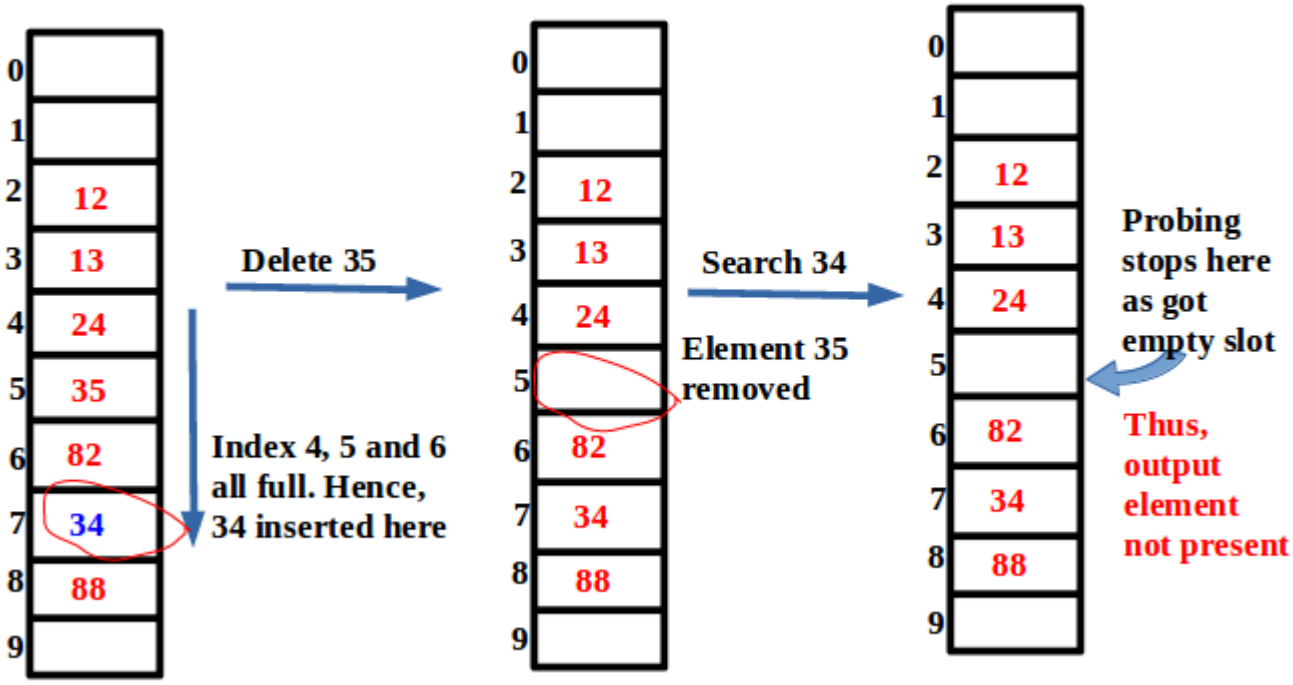
As a group, each team member will write a description of a hashing algorithm. Each group needs to discuss both separate chaining and open addressing techniques used in managing collisions in a hash table. The description need include the heuristic of the algorithm, a provided code example and a cost and benefit analysis of the algorithm and the Big-O run time index of the algorithm.

Algorithms to Choose From:

- Chaining



- Open Addressing



Lab 8 Project Team Assignments

- Lab 8 Groups

Lab 8 Peer and Self Evaluation (https://ohlone.instructure.com/courses/18987/files/3097847/download?download_frd=1)

Testing Specification

Submit a test run validation to demonstrate the hash algorithm.

What to Turn In

Deliverables: **No zip files.**

- .h header file
- .cpp source implementation file
- .cpp test driver file
- Written description of your assigned algorithm
- Evaluation of self/ team member contribution

Your responses to the required lab 8 discussion questions.

Tips and Requirements

1. Ensure that your solution is well organized. Provide a program header and comments to document and organize your source code. User defined function(s) need be documented.
2. Provide a commented out copy of your program run. Enclose the run inside of multi-line comment delimiters so that your program will run in the grader test bed. Place the run after your program source code in the a8.cpp file.
3. Title your submission files in the format ***firstinitalsecondinital***.filename. (Example: **A**nn **O**hlone would submit files **ao**a8.xxx).
4. Answer the questions in the **Lab 8 Required Discussion** thread.

Submission Resources

For more information on how to submit your assignment, please visit:

- **How do I submit an online assignment? Canvas Student Guide (https://community.canvaslms.com/docs/DOC-9539)**
- **Assignments Overview Canvas Video Guide (https://community.canvaslms.com/videos/1122-assignments-overview-students)**
- **Assignments Submissions Canvas Video Guide (https://community.canvaslms.com/videos/1121-assignment-submissions-students)**

Submitting multiple files to an assignment

Your lab 8 assignment requires uploading more than one file; you should upload these multiple files as one submission. To add these files, the **Add Another File button** is clicked to **upload the two files one by one. Check to make sure that both files uploaded okay.** When finished click **Submit Assignment.**

Questions?

Feel free to **ask i**n the forum!

Lab 8 Rubric

Criteria	Ratings			Pts
On time submission	**4 pts** **On time submission**	**2 pts** **One day late**	**0 pts** **Two days late**	4 pts
hash source solution Satisfies the specification requirements to implement either a chained or open addressing hash implementation.	**5 pts** **Full Marks**	**0 pts** **No Marks**		5 pts
Test driver file a8.cpp : Satisfies the test run validation source statements to demonstrate hash function implementation.	**3 pts** **Full Marks**	**0 pts** **No Marks**		3 pts
Hash Algorithm Write-Up Provides algorithm description including heuristics and trade-offs in implementation. Big O runtime provided.	**2 pts** **Full Marks**	**0 pts** **No Marks**		2 pts
Commented out test run included A copy of the program test validation run output is attached after the source code in the main.cpp test driver file (enclosed within comment delimiters). The run matches the source code submitted.	**3 pts** **Full Marks**	**0 pts** **No Marks**		3 pts
Coding style Includes a program header that describes the application. Code is correctly formatted (i.e. follows code style rules). Class methods are documented. Named constants are defined and used in lieu of literal values.	**2 pts** **Full Marks**	**0 pts** **No Marks**		2 pts
Peer and Self Evaluation Feedback Peer and Self Evaluation feedback forum is provided.	**1 pts** **Full Marks**	**0 pts** **No Marks**		1 pts
.h header file Header file to provide user interface to application implementation file. Includes name guard.	**2 pts** **Full Marks**	**0 pts** **No Marks**		2 pts
				Total Points: 22