The Data Structures Project 3

Outline for CS316 'Project Hash'

Project:

This next project is called Project Hash. Refer to project 2 for any details about these class projects, and the theme is test and measure.

Deliverables:

Same deliverables that includes a compiled research document and C ++ code

Repeating the requirements:

1) *A research paper (use the provided example as your guide). Research papers 10 pages (no code);*
2) *Supporting code that works (runs) and supports your research (using your programming skill in C++).*

Input and Output:

Standard C++ programs can be invoked using any IDE or the program may be invoked from the command-line. The inputs are user defined test cases and the output for this project are the results which eventually determine what is empirically true about the data structure. Write this code and maintain a separate analysis with your results from your experiments. Simulate different loads, different inputs/outputs, different conditions, etc., and form a hypothesis. As seen in the example research document, the analysis can be formatted in any number of ways.

A project may not be graded

You must have included one other experimenter, provided true, runtime and space analytics found in our materials regarding hashing and your paper must be 10 fully qualified pages. (Paper is 10 pages and does not include my code.)

How do I know that my project will be graded?

Did you:

- Include one other experimenter?
- Provide true, runtime and space analytics found in our materials regarding hashing?

- Submit 10 pages that did not include any code examples?
- Provide working code?

Test case requirement:

What are we testing? See the lists below for your experiment assignment. If '*assigned to'* has your number listed then that is the structure you will be undertaking for this project. You will write the supporting code and perform the research paper for the structure that is assigned. You will notice some of these structures are variants. The type of data is going to be the choice of the experimenter.

IMPORTANT: For you to receive a grade for this project you must:

1) Provide at minimum one experimenter your code and visa versa. Please provide the code with your code that produces the results.
2) Use time and space complexity formulas to explain your findings.
3) Paper should be 10 pages and this effort do not include code examples in the research doc.
4) Working code that is yours and your partners.

Without one other experimenter the paper will not be graded. Without providing true, runtime and space analytics found in our materials regarding hashing the paper will not be graded. Paper is 10 pages and does not include my code.

Matchup chart:

PROJECT 3

	Assigned to	Experiment with
H1 - Separate Chaining with Link Lists	4,13,22	Anyone
H2 - Separate Chaining with Array Hash Table using a dynamic array	5,14,23	Anyone
H3 - Linear Probing	6,15,24,29	Anyone
H4 - Quadratic Probing	3,7,16,25	Anyone
H5 - Double hashing	12,8,17,26	Anyone
H6 - Cuckoo hashing	20,9,18,27	Anyone
H7 - Hopscotch hashing	1,10,19,28	Anyone
H8 - Two-choice hashing	2,11,20	Anyone

The list of experimenters can be found in the project folder, aptly named 'list of experimenters'. Remember, there intentionally are more possible experimenters listed above and we only need ONE other test instances.

Note for your project to receive a grade, you must find a partner experimenter as mentioned above.

Test case ideas:

Same as before

How do I get my project to be graded?

Include one other experimenter. Provide true, runtime and space analytics found in our materials regarding hashing. Paper must be 10 fully qualified pages. (Paper is 10 pages and does not include my code examples.)

Sourcecode:

You must create your own source code that is used as the focus or basis of the test and subsequent hypothesis. Combine that with your partners and submit for code credit.

Rubric

Research	**75 points**
The format of the work follows the specified guidelines. The hypothesis is well formed. The information regarding the data structure of this research is solid and articulate. The project implemented well considered test cases; thorough and solid testing. The results and measures from the experiments are well outlined and well documented. The analysis is thorough and accurate, and uses good techniques and proofs for considering both the data structure's space complexity and the runtime complexity of each structure overall. The resulting final analysis compares and concludes, with proof, what is optimally efficient (worse case), and supports the hypothesis.	**75**

Coding and Accuracy	**25 points**
Program is properly documented, such as header and comments, purpose, and instructions on how to use your program. Implementations follow from design patterns. The program is well designed, logically sound (no segment faults), produces correct results (eg., correctness after insertion, removal, etc…). Program solution code does not use the STL for its data structure. IMPORTANT: Program properly handles memory (no memory leaks). The behavior of the code maintains the properties of the data structure accurately and predictably. For example, the properties of a BST or max-heap and so on, are not violated, no duplicates allowed where applicable, performs all of the basic operations of an AVL tree or linked list or handles all stack operations or probing functions, etc…	**25**

Total =	**100**

How do I know that my project will be graded?

Did you:

- Include one other experimenter?
- Provide true, runtime and space analytics found in our materials regarding hashing?
- Submit 10 pages that did not include any code examples?
- Provide working code?

Summary/notes:

- Your program should compile and run;
- Your program may not use anything from the Standard Template Library for your data structure;
- Your program should run tests against ONE other experimenters;
- Make sure your program can pass the test cases before submitting the code.

What code do I submit? You **will** need to **submit** another experimenter's code with yours, and it must produce the outcomes found in your report.

Submission Instructions – for programming solutions

On Brightspace, go to the matching folder for the Project ??, where ?? is the project name of the project that is assigned (eg., Project Treeeeps for the second project), and submit the appropriate documents, which includes any code (cpp) and header files (hpp), and the project analysis (doc, docx) that is at least **10 pages**. So, your program code and the report (document) is all you will submit.

How do I know that my project will be graded?

Did you:

- Include one other experimenter?
- Provide true, runtime and space analytics found in our materials regarding hashing?
- Submit 10 pages that did not include any code examples?
- Provide working code?