

# CS 124 Project 5

Due by 11:59pm on Friday, December 3rd.

Ten percent extra credit will be given to projects submitted by Friday, November 19th.

No early submission extra credit will be given to projects submitted between 11/20 and 12/2.

For this project, you will hash the 1000+ objects from your data set. You will experiment to see what size hash table, collision detection algorithm, hash function, and key work best for your data.

---

## Implement

- You should have your 1000+ objects stored in a vector.
- Insert your objects into two hash tables:
  - 1) One with separate chaining.
  - 2) One with non-linear probing. You can choose quadratic probing, exponential probing, or double hashing. I recommend that you make a copy of the LinearProbing class and modify it.
- Calculations for each hash table:
  - a) Record the number of hash elements you look at (read) for each insertion. What is the maximum and average number of reads?
  - b) Create four more hash tables of varying sizes (all of which should be greater than the size of your data set) and see what happens to the number of reads you record. Your program should create five separate chaining and five probing hash tables. For the probing hash tables, what is the size after inserting all of your elements? Which hash table size is best for the data set?
  - c) For separate chaining and probing, create a hash table with a different hash function, a hash table with a different getKey function, and a hash table with both functions changed. You should use the table size that is best for your data set, as determined in the previous part. See what happens to the numbers of reads you record. What is the effect on the placement of elements in the hash table? Which hash/getKey combination is best for your data set?
  - d) Consider how removing and searching compares with inserting, in terms of number of hash elements read. Would they read more, less, or the same?
  - e) Include graphs of your read data and answers to all of the above questions in your report.
- How often would each of the three operations (inserting, removing, searching) typically be used with your data set? Which hash collision method works best for your data set? Why? Include answers to these questions in your report.
- You must submit your .cpp and header file(s), your data file(s), and your report. Your source files should include file output of all the read counts collected from the prompts above. Your report must be submitted in PDF format.

---

## Extra Credit

For up to 10 points extra credit (at the grader's discretion), you can:

- Use timers to see how long it takes you to insert/find elements in the hash tables.
- Compare those times with the time it takes to insert/find elements stored in other data structures (e.g. an unsorted vector, a sorted vector, a binary search tree, an AVL tree, a splay tree, a heap, etc.) The more structures you include, the better!

---

## Grading

The project is out of 80 points.

- 5 pts Program compiles and runs.
- 5 pts Code style. Readable, naming style is consistent, comments where appropriate.
- 5 pts Hashed at least 1000 objects using separate chaining.
- 15 pts Hashed at least 1000 objects using non-linear probing.
- 5 pts Used at least 5 different hash table sizes, as specified above.
- 15 pts Used two hash functions and getKey functions, as specified above.
- 10 pts Recorded the number of reads for each type of hashing.
- 15 pts Analyzed the results and wrote about everything outlined above.
- 5 pts Report: professional, grammatically correct, cites sources.