

Employee Database – Part 2



Performance tests

You will implement performance tests below in new clients, **TesterLinearP.java** and **TesterQuadP.java** (performance test). Simply coding a hash table is "good enough" at this point, but *understanding* why and how hash tables work, and the trade-offs in memory / performance is important for an aspiring computer scientist.

You will use the data in the "**Large Data Set.txt**" file that contains names and IDs. You will also use the files "**Successful Search Records.txt**" that contain entries that exist in the original data set to test the performance of your hash table (make sure you look at the files so that you understand the format of the input). You will perform the following tests for both the linear and quadratic schemes:

1. Read the contents of the text files into your program, store in a convenient data structure.
2. You will build 3 different tables for different values for α (where load factor α is number of entries per number of buckets – not sure what is a load factor? – Google my friend).
 - α : 0.10, 0.50, 0.99

Given the input file of 50k records, build a table in **EmployeeDatabase** sized for the load factor to be profiled.

- a. e.g. for an $\alpha = 0.50$, the table size should be ~100k
 - b. e.g. for an $\alpha = 0.67$, the table size should be ~75k
3. Start "Build Table" timer //the `System` class contains a method that will return the current time
 4. For each record in the "Large Data Set.txt":
 - a. Parse the record, build an object, insert it into the hash table
 5. Stop "Build Table" timer
 6. Start "Successful Search" timer
 7. For each record in "Successful Search" data set:
 - a. Parse the record
 - b. Search the table – get number of probes needed to find the entry
 8. Stop "Successful Search" timer

9. Output a report containing:

- a. Hash function used (your descriptive designation)
- b. Number of records added to the table, and table size at each load factor
- c. Average insertion time at each α (load factor)
- d. Average get time at each α (load factor)

NOTE: Records in the provided data files have unique names – if you get a name match, you don't need to check the ID to be sure it's the correct record (if you were selling this program, you would have to check!).