Matt Clyde

Phil Tseng

Shawn Forgie

**CS350 Project Proposal – Dictionaries**

For our project we intend to create various implementations of dictionaries and gauge their relative efficiency. We decided on this project because it allows for a wider variety of data structures than the others suggested and is familiar to all of us from our use of the built-in Map and Dictionary classes found in many languages. Coding will be done in Groovy using the Intellij IDE, with Github as source control. The reason for this is the three of us have worked together on projects in the past using the same IDE and source control along with the Java programming language, and Groovy is largely an extension of Java with a few extra beneficial features. We will primarily communicate via Skype sessions with screen sharing, with additional face-to-face meetings directly following Tuesday and Thursday class sessions as needed.

At this time we intend to develop three separate dictionary implementations: a hash table with chaining using arrays of linked lists, a binary search tree and a 2-3-4 tree, with the hash table run using several different hashing functions. For each, our report will include the following:

* Description of the search algorithm for each implementation
* Steps of each search algorithm, in pseudocode
* Binary Search Tree
* 2-3-4 Tree
* Hash Table with simple hash function
* Hash Table with cryptographic hash function
* Explanation of the testing methods used for each
* Three sets of key/value test data
* 50,000 English words / definitions (string/string)
* ~1000 GPS coordinates / location names (tuple/string)
* 1,000,000 Random strings / random strings (string/string)
* For each set
* Insert data in random order
* Perform 1000 searches
* Re-insert data in random order and repeat
* Types of keys to use as search parameters in each test set
* Random strings
* Words from English texts
* Words from foreign-language texts
* GPS coordinates from list of known locations
* Randomly generated GPS coordinates
* All tests run on at least two different computers to account for hardware and operating system differences
* Results of those tests
* Means and medians for wall clock time and operation counts for each data structure
* Means and medians for the above for each type of input data
* Graphs of relative efficiency of each implementation on data of varying sizes
* Analysis of those results (esp. with regard to complexity theory):
* Which data structure was most efficient for searching?
* How much of an effect did the input data types have?
* How much of an effect did the query types have?
* How long did searches typically take?
* What was the efficiency class of each data structure?
* What further insights and conclusions were reached from this assignment?

Below is a proposed schedule for the time remaining in the course:

Week 1 (Feb. 16 – 21):

* Set up source control and IDEs
* Plan, and begin writing code for hash table implementation
* Outline tests and methods of generating test data
* Begin collecting/generating test data

Week 2 (Feb. 22 – 28):

* Finish coding and debug hash table
* Develop and run tests for the hash table
* Plan other two implementations

Week 3 (Mar. 1 – 7):

* Write and debug code for trees
* Develop and run tests for trees
* Validate results of all tests

Week 4 (Mar. 8 – 13):

* Collect and analyze test results
* Create graphs to present results for the final report
* Write and submit analysis

It is our intention to work together on the planning and design of each implementation, as well as on the test cases and data generation methods. The coding of the dictionaries themselves will be divided among team members, with code reviews to be performed by the others once a stable build has been developed. Collection and analysis of our results will be an ongoing endeavor, with contributions to a tracked data store on completion of each test run. All team members will run all tests on their own machines in order to track potential effects of hardware. The final paper submission will be created by the group as a whole during one or multiple face-to-face sessions.