David Johnston

Professor Wei Kian Chen

CSI-281, Data Structures and Algorithms

7 October 2013

Programming Assignment 7 Report

For this assignment I the data type which I ultimately decided to use is a hash table. I decided this for a variety of reasons. First and foremost, hash tables interested me for their potential to be incredibly fast. Speed was not the only consideration, I also thought that a hash table would be really fun to mess around with and not too difficult to implement.

I ultimately decided on an implementation of the FNV-a1 hash algorithm, with xor-folding down to 23 bits. FNV-a1 seemed to provide the greatest compromise between speed, uniqueness, and ease of implementation. The xor-folding was necessary to reduce the potential size of the hash value so that I could actually allocate an array for the keys. To handle collisions I ended up using a linear probing technique. This simply was the first thing to come to mind as a simple, quick way to handle collisions. It was easy to implement, and, as the greatest offset from the original hash position that occurs given my hash implementation and the provided codebook is only three, does not reduce the speed of the algorithm to a great extent.

I would say overall the hash table was a great choice. It was relatively simple to implement and it is very quick. I would love to play around with different hash algorithms in the future. I don’t feel anything negative about FNV-1a really; I just would like to try out a bunch of other ones. I think if I were to do the assignment again, I would use some other method of collision handling. My current method works well enough, but I feel like the abundance of string comparisons must be very slow.