Cullen Williams, Garrick Hutcheson
Hassan Ghasemzadeh
CPTS 223
26 April 2018

Homework 10 report

Problem Statement:

The goal of this assignment is to compare the Insertion and search performance of different hashing functions. The first part of this project is designed to illustrate the differences between three collision handling strategies: chaining, linear probing, and quadratic probing. The second part of this project revolves around how different hashing algorithms affect data distribution, and as a result of this the overall number of collisions.

Experimental Setup:

Machine specification :
Acer v5-131
OS: Ubuntu
Ram: 5.7 GiB
CPU: Intel Celeron 1007U @ 1.50GHz x2
Compiler: Terminal with G++

Both Aim1 and Aim2 were run ten times and the results were averaged to obtain the data points in the tables below. Test results were fairly consistent and always yielded the same number of collisions and unsuccessful probes, as would be expected.

Experimental Results:

Aim 1:

Collision Strategy	Insert			Search	
	Total time (µs)	Average time (µs)	Total number of collisions	Total time (µs)	Average time (µs)
Chaining	46141	13	3266	900	0.5996
Linear Probing	37256	10.6568	3073	651	0.433711
Quadratic Probing	30362	8.68478	3074	657	0.437708

Aim 2:

Collision Strategy	Insert			Search	
	Total time (µs)	Average time (µs)	Total number of collisions	Total time (µs)	Average time (µs)
Simple	52180	14.9256	11146	3157	2.10326
Prefix	31481	8.99457	8354	886	0.590273
Full-length	32287	9.23541	3220	975	0.649567

Despite the input arrays being the same for Full-Length Hashing and std::hash the number of collisions and overall performance of std::hash was significantly lower than that of Full-Length Hashing. This implies that Full-Length Hashing function from figure 5.4 and std::hash are not the same function.