Lab 5 Report

1. Observe your output for quadratic probing and double hashing and compare the performance of both techniques for the build and find operations.

Based on how the program has been set up for this lab, the hash table with double hashing often takes longer to build than the hashing table with quadratic probing for each input size into the tables.

2. Which hashing technique performs better when searching and why?

The quadratic probing table appears to be more successful at finding table items than the double hashed table. Where the double hashed table's rate of success peaks at below 8%, the quadratic probing table's rate of success peaks out at below 10% and was able to substantially find more items successfully at every point of input size than its counterpart. Perhaps it is because the quadratic probing table distributes the randomly generated numbers into the empty buckets more evenly throughout the table to allow for a wider inspection for an item to find that results in the quadratic protable's substantive advantage. The double hashing table afterall, while not prone to primary clustering compared to a linear probed table, is more susceptible to primary clustering than the quadratic probing table because of its non-exponential probing.

3. Justify the worst case complexity of each of your experimental profiling results and compare them to the worst case complexity of the theoretical results for the build times with each of the hashing techniques.

In the experimental worse case for both tables, the input size is equal to half the length of the tables - 1. Because the input size is less than half the length of the table, this ensures that the load factor will always remain below 0.5, and thus no rehashing ever occurs.

In the theoretical worst case for both tables, the input size is equal to the table length before the random inputs are inserted. Assuming that both tables rehash when the load factor is greater than 0.5, both tables would be rehashed and would cost no more than double the initial build time of the table before the rehash in order to double the table length and reinsert every previously inserted input into their original indexes. Continuing from there, the lower half of the remaining inputs after the rehashing would then have to be inserted, and the build time for both tables be finalized after all inputs are inserted if possible.

Because the experimental worse cases never result in a rehash, any theoretical worse cases could result in exponentially greater build_times than the experimental worse cases, even at the base case of an input_size equal to the table length.