Austin Johnson

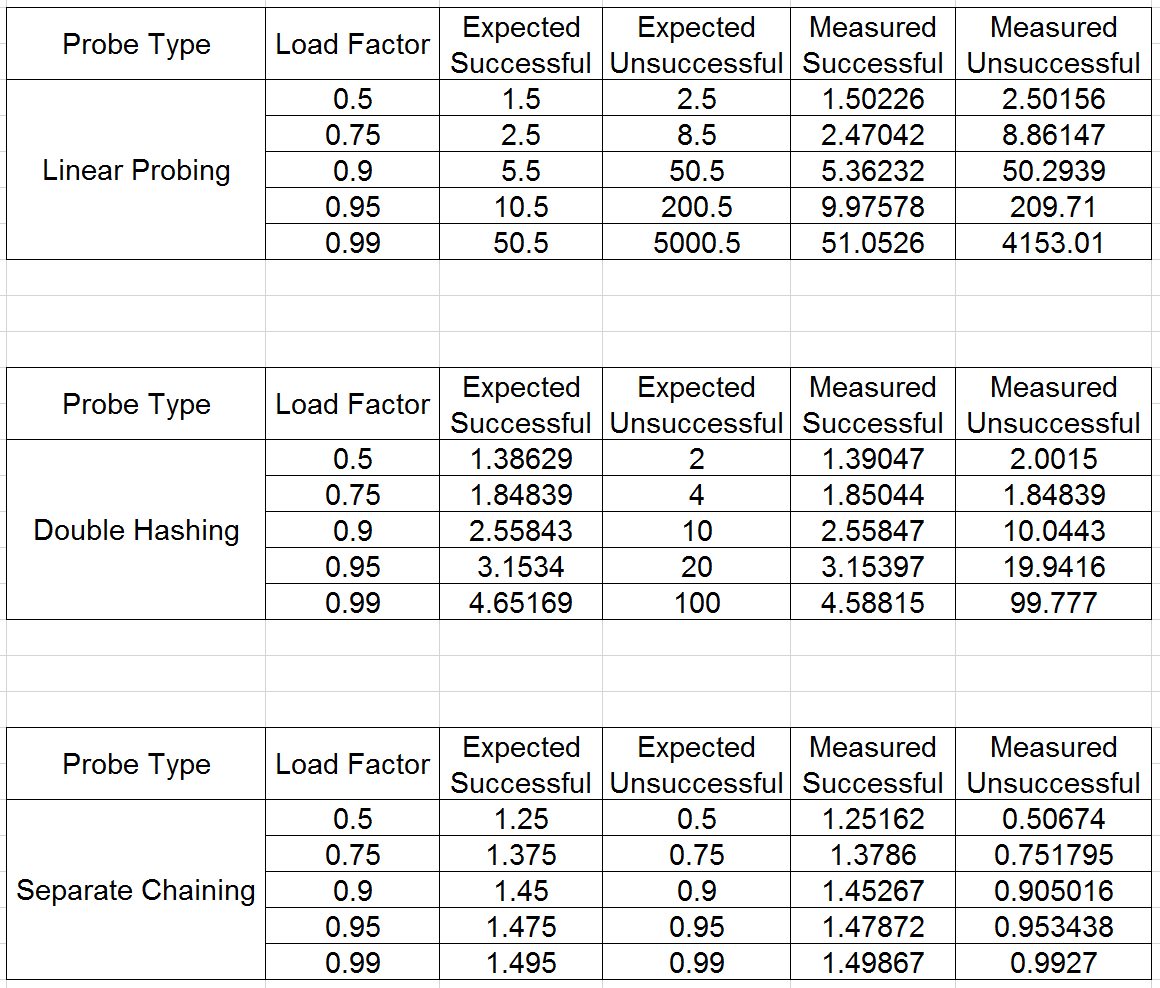
ECE 2230

MP7: Hash Table ADT

3 December 2016

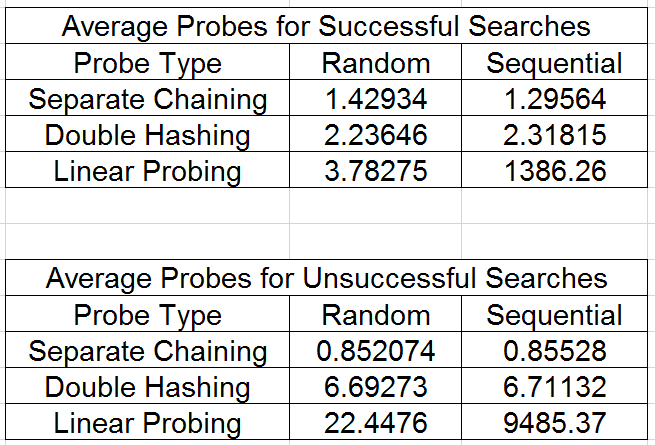
Performance Analysis

1. Comparing expected and measured probes:



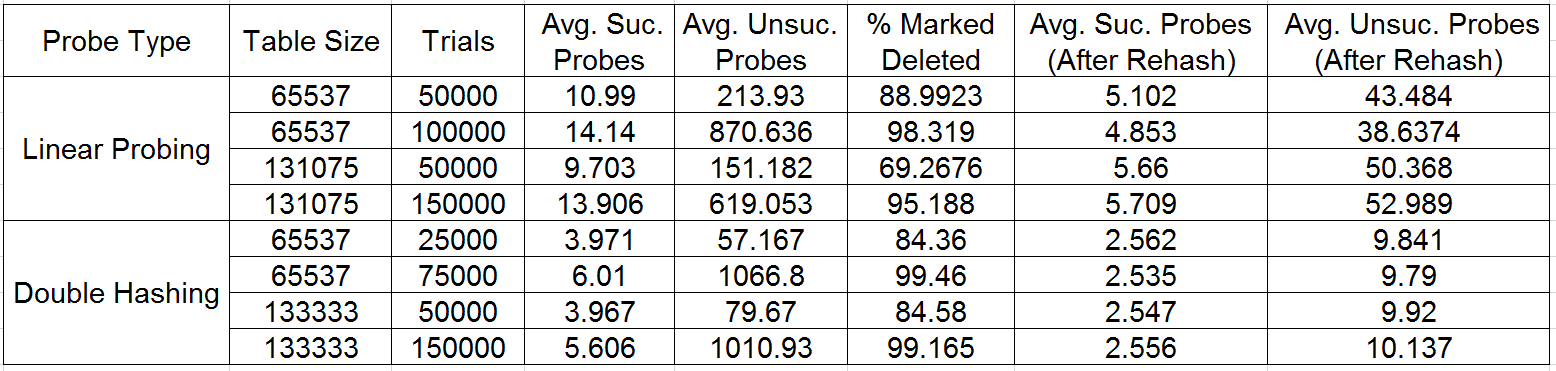
The number of probes I measured when calling retrieve were relatively close to the expected values from the performance equations. For linear probing as the load factor of the table increases, the more the measured unsuccessful probes vary from the expected unsuccessful probes. For all other cases, the measured and expected values are very close to one another. Any discrepancies between the two values are most likely due to the hash function being used. The expected values assume a perfect hash function is being used, and while the hash function I implemented performed well, it is not perfect.

2. Approximating the time complexity for each probe type:



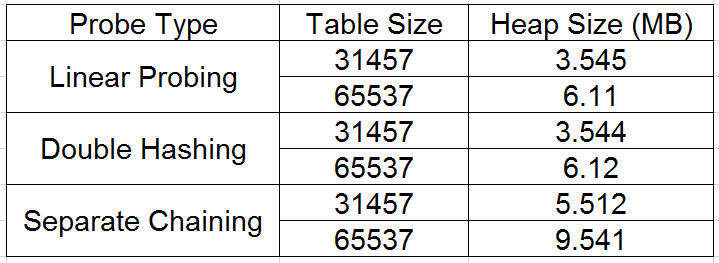
For open addressing when using random inserts, the results suggest that it is roughly O(1) search time for both successful and unsuccessful searches. When the inserts are sequential, the results suggest that it is still roughly O(1) for double hashing but it is O(n) for linear probing. For separate chaining it is always O(1) regardless of the way it was inserted and if it was a successful search or unsuccessful search.

3. Proving that an increased table / trial size delivers worse performance:



As the amount of entries in the table marked deleted increases, the amount of probes required to search increases. We can see that once the table is rehashed, the amount of probes required to search is more comparable to the expected amount of probes required to search.

4. Comparison of the memory usage by each probe type:



Separate chaining has a lot more memory usage over open addressing due to the array type being used. Separate chaining uses an array of pointers to structures. Each time we insert using separate chaining, we have to dynamically allocate memory for that structure.