# Performance Analysis – MP7

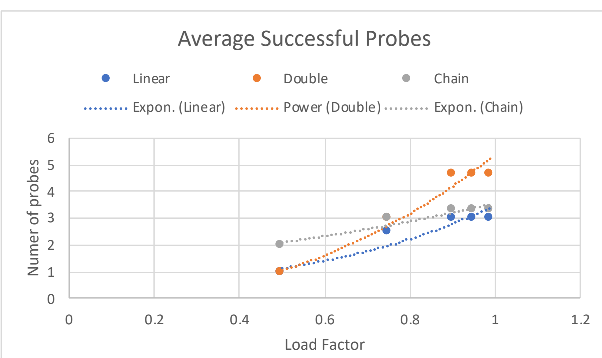
Cameron Burroughs
ECE 2230
Dr. Russel

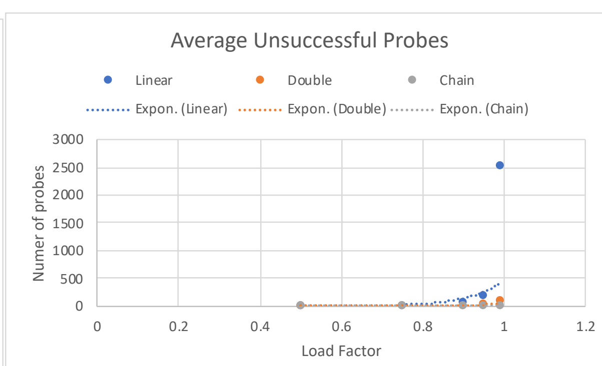
Part 1:

| Actual Results | | Load Factor | | | | |
|---|---|---|---|---|---|---|
| | | 0.5 | 0.75 | 0.9 | 0.95 | 0.99 |
| **Linear Probing** | avg. successful | 1 | 2.5 | 3 | 3 | 3 |
| | avg. unsuccessful | 2.52 | 8.57 | 48.9 | 164.1 | 2526 |
| **Double Hashing** | avg. Successful | 1 | 3 | 4.66 | 4.66 | 4.66 |
| | avg. unsuccessful | 1.99 | 4 | 10 | 19.99 | 98.99 |
| **Separate Chaining** | avg. Successful | 2 | 3 | 3.333 | 3.333 | 3.333 |
| | avg. unsuccessful | 1.5 | 1.75 | 1.9 | 1.95 | 1.95 |

| Expected Results | | Load Factor | | | | |
|---|---|---|---|---|---|---|
| | | 0.5 | 0.75 | 0.9 | 0.95 | 0.99 |
| **Linear Probing** | avg. successful | 1.5 | 2.5 | 5.5 | 10.5 | 50.5 |
| | avg. unsuccessful | 2.5 | 8.5 | 50.5 | 200.5 | 5000.5 |
| **Double Hashing** | avg. Successful | 1.84 | 1.84 | 2.55 | 3.15 | 4.65 |
| | avg. unsuccessful | 4 | 4 | 10 | 20 | 100 |
| **Separate Chaining** | avg. Successful | 1.25 | 1.37 | 1.45 | 1.47 | 1.495 |
| | avg. unsuccessful | 0.5 | 0.75 | 0.9 | 0.95 | 0.99 |

Generally, Linear probing was more accurate for smaller load factors, whereas double hashing was more accurate for load factors of around 0.9 and up. Separate chaining was probably the least accurate, almost always being 2 times the expected average probes. This could be due to inefficient coding, or an incorrect probe increment, however it still reflects close to O[1] performance, as the avg successful and unsuccessful probes have little difference as the load factors increase.




Part 2:

Average Successful Probes

| | Random | Sequential |
|---|---|---|
| Separate Chaining | 3 | 1 |
| Double | 3 | 2.21 |
| Linear | 2.5 | 1729 |

Average Unsuccessful Probes

| | Random | Sequential |
|---|---|---|
| Separate Chaining | 1.85 | 1.8 |
| Double | 6.66 | 6.62 |
| Linear | 21.6 | 8731 |

Separate chaining and double have little difference in performance based on the type of list created, however linear probing performs significantly worse with a large sequential list, suggesting O(n) performance.

Part 3:

50,000 Trials

| | | Performance before rehash | | Performance after rehash | |
|---|---|---|---|---|---|
| | % deleted | successful | unsuccessful | successful | unsuccessful |
| Linear Probing | 89.65 | 11.8 | 290 | 5.3 | 46.24 |
| Double Hashing | 97.87 | 5.1 | 471 | 2.5 | 10 |
| Separate chaining | 0 | 2.58 | 1.9 | 2.58 | 1.9 |

100,000 trials

| | | Performance before rehash | | Performance after rehash | |
|---|---|---|---|---|---|
| | % deleted | successful | unsuccessful | successful | unsuccessful |
| Linear Probing | 98.64 | 15.5 | 1581 | 5.74 | 55.7 |
| Double Hashing | 99.97 | 6.8 | 21096 | 2.5 | 9.77 |
| Separate chaining | 0 | 2.65 | 1.9 | 2.65 | 1.9 |

With more trials, the percentage of entries marked deleted greatly increases, causing the number of probes for insertion to increase in open addressing type lists. For linear and double hashing probe types, the average number of probes for an unsuccessful search was fairly large at 50,000 trials and even larger at 100,000 trials. After rehashing the table, however, performance went back down to what was expected. This is because “deleted” entries cause the list to be completely searched to ensure there is no duplicate entry. This is mainly what causes O(n) performance. For separate chaining, the performance was identical not only before and after rehashing the table, but also with an extremely large number of trials. This suggests O(1) performance.

Part 4:

| | Heap size |
|---|---|
| Linear Probing | 431,648 bytes |
| Double Hashing | 431,648 bytes |
| Separate chaining | 431,688 bytes |

Generally, the trade-off of using separate chaining for better performance means more memory is required. For my specific implementation, with the same number of items added to the list, separate chaining did not use much more memory than the open addressing methods, this is most likely due to the fact that a sep_chain_t type is only allocated when there is something to be inserted into the list, whereas open addressing types are all allocated when the list is made.