HW6 Hashing & Symbol Tables

1. Hash Results

|  |  |
| --- | --- |
| inputs | h(k) = k % 11 |
| 4371 | 4 |
| 1323 | 3 |
| 6173 | 2 |
| 4199 | 8 |
| 4344 | 10 |
| 9679 | 10 |
| 1989 | 9 |

* 1. Separate Chaining - Linear Probing

9679

|  |  |
| --- | --- |
| 0 |  |
| 1 |  |
| 2 | 6173 |
| 3 | 1323 |
| 4 | 4371 |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 | 4199 |
| 9 | 1989 |
| 10 | 4344 |

|  |  |
| --- | --- |
| 0 | 9679 |
| 1 |  |
| 2 | 6173 |
| 3 | 1323 |
| 4 | 4371 |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 | 4199 |
| 9 | 1989 |
| 10 | 4344 |

* 1. Secondary Hash Function –

|  |  |
| --- | --- |
| 0 |  |
| 1 | 9679 |
| 2 | 6173 |
| 3 | 1323 |
| 4 | 4371 |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 | 4199 |
| 9 | 1989 |
| 10 | 4344 |

1. Advantages and disadvantages of different hash strategies
   1. Separate chaining –
      * Disadvantages
        + For searches we might end up having to linear search a long list.
        + Potential for wasted space in table.
      * Advantages
        + We avoid entering a loop for placing an item into a hash table.
        + We can keep lists short (array resizing is optional)
   2. Linear Probing-
      * Disadvantages
        + Potential for an infinite loop if there are no available spaces.
        + Placing could become excessively long.
        + Searches too could become long.
      * Advantages
        + Space is used efficiently, no wasted table space.
        + Potentially faster because of locality in memory. (no ll refs)
   3. Secondary Hash-
      * Advantages –
        + Basically has the advantages of linear probing but without the downside of having to potentially do a long linear search.
        + Less collisions
      * Disadvantages –
        + You have to find a good second hash function to make it worth it.
2. My hash function – h(k) = k%47

|  |  |  |
| --- | --- | --- |
| Key | h(k) | Hash Location |
| 100 | 6 | 6 |
| 0 | 0 | 0 |
| 1199 | 24 | 24 |
| 1299 | 30 | 30 |
| 1399 | 36 | 36 |
| 2 | 2 | 2 |



|  |  |  |  |
| --- | --- | --- | --- |
| Key | h(k) | h2(k) | Hash Location |
| 100 | 64 |  | 64 |
| 0 | 0 |  | 0 |
| 1199 | 24 |  | 24 |
| 1299 | 88 |  | 88 |
| 1399 | 51 |  | 51 |
| 2 | 74 |  | 74 |

1. Load factor = 180/200 = 0.9 = α
2. Linear Probing – 1/2(1 + 1/1-α) = ½(1 + 1/1-0.9) = 5.5

Separate Chaining – (1 + α/2) = 1.45

Double Hashing- (-ln(1 -α))/α = 2.55842

1. 1k elements need to go into a hash table…
   1. Linear Probing – ½(1+ 1/1-α) = 2

1+ 1/1-α = 4

1/1-α = 3

1. = 3(1-α)

1 - 1/3 = α = 2/3

1000/1500 = 2/3

1500 = array size

* Separate Chaining – 1 + α/2 = 2

α = 2

1000/500 = 2

500 = array size

* Double Hashing - (-ln(1 -α))/α = 2

array size = 1255



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| keys = 10k | 11,000 | 15,707 | 17,111 | 25,111 |
| unsuccessful | 6 | 1.87 | 1.70 | 1.33 |
| successful | 3.38 | 1.34 | 1.26 | 1.09 |