Zijian Gao
26319560
2/29/2016
CS311 HW2

11.2-3.)

For this problem I assume that the chained lists are sorted by key.

For successful searches, the fact that each chained list is in sorted order would give a worst-case runtime of $\theta(\log\alpha)$, including some constant factor for the hashing function. This is because a binary search can always be used to locate the element in the list. The best-case runtime is $\theta(1)$, if the element is located at the head or tail of the chained list.

For unsuccessful searches, the worst-case runtime would be $\theta(2)$, as the first step in a search would be to test if the desired key falls between the minimum and maximum key value within each chained list, which requires a constant two iterations to complete. The best-case is $\theta(1)$ as this would be the case where the key does not hash to any populated slot.

For insertions, the worst-case runtime would be $\theta(\alpha\log\alpha)$, as the load factor gives the average number of elements per chain, and the worst-case time to sort that amount of elements would be $\theta(\alpha\log\alpha)$ (merge-sort or similar). The best-case runtime would be $\theta(1)$, where the key for the element that hashes to a certain chain is either the greatest or minimum key compared to the rest of the keys in the chain and only appending to head/tail of the list is required.

For deletions, since deletions rely on searches, the runtime is proportional to the runtime to find the target element for deletion, $\theta(\alpha\log\alpha)$. The only additional work required is to keep track of the previous item in the linked list, and point that element to the item after the deleted item to properly delete the item, and this is a constant factor.

11.4-1.)

Keyset: {10,22,31,4,15,28,17,88,59 }
Keyset(mod 11): {10, 0, 9, 4, 4, 6, 6, 0, 4}
Using linear probing, first key 10 is inserted into slot 10. Then key 22 is inserted into slot 0 since $h(k,i)=(k+i)\bmod 11$. Then, key 31 is inserted into slot 9. Key 4 is inserted into slot 4. The first collision occurs at key 15, which also maps to slot 4. The hash function would then probe the next slot, 5, which is empty, so key 15 is inserted into slot 5. Next, key 28 is inserted into slot 6, which is empty. Key 17 also hashes to slot 6 at first, so linear probing finds that slot 7 is empty, and key 17 is inserted into slot 7. Then, key 88 collides with key 22 stored at slot 0, so the next empty spot, 1, is used and key 88 is inserted into slot 1. Finally, key 59 collides with keys 4,15,28,17 which are stored at slots 4,5,6,7 respectively, so probing finds the next (i = 4) slot, 8, which is empty and key 59 is stored into slot 8. So, in the end, slots 0,1,4,5,6,7,8,9,10 are occupied.

With quadratic probing where c_1 is 1 and c_2 is 3, $h(k,i)=(k+i+3i^2)\bmod 11$. Here, key 10 is inserted into slot 10, which is empty so probing does not need to be done. Key 22 is inserted to slot 0, then key 31 to slot 9, then key 4 to slot 4. The first collision occurs at

Key 15, which is inserted into $h(15,1) = (15+1+3) \bmod 11 = 8$ slot 8. Key 28 is mapped to slot 6. Key 17 collides with key 28, and is mapped to:

$h(28,1) = (28+1+3) \bmod 11 = 10$

$h(28,2) = (28+2+12) \bmod 11 = 9$

$h(28,3) = (28+3+27) \bmod 11 = 3$

Slot 3, which is the next empty spot.

Key 88 collides with key 22, and is mapped to:

$h(88,1) = (88+1+3) \bmod 11 = 4$

$h(88,2) = (88+2+12) \bmod 11 = 3$

$h(28,3) = (88+3+27) \bmod 11 = 8$

$h(28,4) = (88+4+48) \bmod 11 = 8$

$h(28,5) = (88+5+75) \bmod 11 = 3$

$h(28,6) = (88+6+108) \bmod 11 = 4$

$h(28,7) = (88+7+147) \bmod 11 = 0$

$h(28,8) = (88+8+192) \bmod 11 = 2$

Slot 2.

Key 59 collides with key 4, and is mapped to:

$h(59,1) = (59+1+3) \bmod 11 = 8$

$h(59,2) = (59+2+12) \bmod 11 = 7$

Slot 7. At this point, all 9 keys are stored into all 9 slots. In order from first to last, keys(first to last in the given order) are inserted into slots 10,0,9,4,8,6,3,2,7.

With double hashing, the hash function is $h(k,i) = (k + i(1 + k \bmod (11-1))) \bmod 11 = (k + i + ik \bmod 10) \bmod 11$. First, key 10 is inserted into slot 10. 22 to slot 0, 31 to slot 9, 4 to slot 4. Key 15 collides with key 4 and is mapped to:

$h(15,1) = (15+1+15 \bmod 10) \bmod 11 = 21 \bmod 11 = 10$

$h(15,2) = (15+2+30 \bmod 10) \bmod 11 = 17 \bmod 11 = 6$

Slot 6, since slot 10 (i=1) is filled.

Key 28 collides with key 15, so it is mapped to:

$h(28,1) = (28+1+28 \bmod 10) \bmod 11 = 4$

$h(28,2) = (28+2+56 \bmod 10) \bmod 11 = 36 \bmod 11 = 3$

Slot 3.

Key 17 also collides with key 15 (slot 6) so it is mapped to:

$h(17,1) = (17+1+17 \bmod 10) \bmod 11 = 25 \bmod 11 = 3$

$h(17,2) = (17+2+34 \bmod 10) \bmod 11 = 23 \bmod 11 = 1$

Slot 1.

Key 88 collides with key 22(slot 0) and is mapped to:

$h(88,1) = (88 + 1 + 88 \bmod 10) \bmod 11 = 97 \bmod 11 = 9$

Slot 9

Key 59 collides with key 4(slot 4) and is mapped to:

$h(59,1) = (59 + 1 + 59 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,2) = (59 + 2 + 118 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,3) = (59 + 3 + 177 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,4) = (59 + 4 + 236 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,5) = (59 + 5 + 295 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,6) = (59 + 6 + 354 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,7) = (59 + 7 + 413 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,8) = (59 + 8 + 472 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,9) = (59 + 9 + 531 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

$h(59,10) = (59 + 10 + 590 \bmod 10) \bmod 11 = 69 \bmod 11 = 3$

So h(k,i) for all i from 0 to m-1 does not give a unique sequence for k=59, and the hash table with double hashing in this case would throw an overflow error as given in the book. In order from first to last, keys in their given order are successfully mapped to slots 10,0,9,4,6,3,1,9.