Week 13 Exercises Part1

Sandra Batista

1. Hash Table Insertion

a) Start with an empty hash table of size m = 7. Use the hash function h(k)=k%7 and draw the contents of a hash table after inserting keys 14, 8, 21, 2, 7 using chaining.

1. Hash Table Insertion

b) Start with an empty hash table of size m = 10. Use the hash function h(k)=k%10 and draw the contents of a hash table after inserting keys 1, 11, 2, 21, 12, 31, 41 using linear probing.

c) Start with an empty hash table of size m = 9. Use the hash function h(k)=k%9 and draw the contents of a hash table after inserting keys 36, 27, 18, 9, 0 using quadratic probing.

- Given the following hash table, trace the steps for find(3). The hash function is h(k)=k%7 (m=7) and quadratic probing was used to hash the items.

- <u>14</u> <u>8</u> <u>21</u> <u>2</u> <u>7</u> __ __
   0 1 2 3 4 5 6

- Given the following hash table, trace the steps for remove(21) followed by find(7). The hash function is h(k)=k%7 (m=7) and linear probing was used to hash the items.

- <u>14</u> <u>8</u> <u>21</u> <u>2</u> <u>7</u> __ __
  0 1 2 3 4 5 6

4. Bloom Filter Exercise

- ## Problem 5 from practice final a:

- ## https://bytes.usc.edu/cs104/resources/final-a.pdf

- We have a Bloom Filter with an array of 10 elements (the elements of the set are integers), and using three hash functions:

- $h1(x) = (7x + 4) \bmod 10$

- $h2(x) = (2x + 1) \bmod 10$

- $h3(x) = (5x + 3) \bmod 10$.

- We execute the sequence of operations given below. What does the program output? Which of the answers are false positives (the Bloom filter says "Yes", even though the correct answer is "No")? Which are false negatives (the Bloom filter says "No", even though the correct answer is "Yes")? If your final answer is incorrect, you may get more partial credit if you show enough work for us to isolate the mistake.

```
BloomFilterSet bf (10);
 bf.add (0);
bf.add (1);
bf.add (2);
bf.add (8);
// Show us what the Bloom Filter's Array looks like at this point.
if (bf.contains (2)) std::cout << "2\n";
if (bf.contains (3)) std::cout << "3\n";
if (bf.contains (4)) std::cout << "4\n";
if (bf.contains (9)) std::cout << "9\n";
```