Kyle Harris – Lab 14 CS 113

Kyle Harris

Lab 14

self-check 7.3

2.

name	hashCode()	hashCode()%7	hashCode()%13
"Tom"	84274	1	8
"Dick"	2128869	0	1
"Harry"	69496448	0	6
"Sam"	82871	6	4
"Pete"	2484038	4	11

Open addressing and linear probing

when size is 7:

1. "Tom" is inserted at [1]
2. "Dick" is inserted at [0]
3. "Harry" is inserted at [6], collides, goes to [1], collides again, then goes to [2].
4. "Sam" is inserted at [6]
5. "Pete is inserted at [4]

when size is 13:

1. "Tom" is inserted at [8]
2. "Dick" is inserted at [1]
3. "Harry" is inserted at [8], collides, then goes to [9]
4. "Sam" is inserted at [4]
5. "Pete" is inserted at [11]

Kyle
Harris

Lab 14 (cont)

Chaining

For size 7:

[0] → "Dick" → "Harry"
[1] → "Tom"
[2] → null
[3] → null
[4] → "Peter"
[5] → null
[6] → "Sam"

For size 13:

[0] → null
[1] → "Dick"
[2] → null
[3] → null
[4] → "Sam"
[5] → null
[6] → null
[7] → null
[8] → "Tom" → "Harry"
[9] → null
[10] → null
[11] → "Peter"
[12] → null

Kayla Harris

Lab 14 (cont)

3. Table length is 5, so using the value with the mod old operation and the length will determine the insertion position.

24 % 5 = 4, [illegible]
6 % 5 = 1, [illegible]
20 % 5 = 0, [illegible]
14 % 5 = 4, [illegible]

1. Insert 6 at [1]
2. Insert 14 at [4]
3. Insert 24 at [4], this collides, so it moves to [0]
4. Insert 20 at [0], this collides at [illegible] there, then shifts to [1], collides again, then finally shifts to [2]

Elements that found current place from collision: 20 and 24

Chaining

[0] → 20
[1] → 6
[2] → null
[3] → null
[4] → 14 → 24