CS 260 Homework 4

Jason Gallagher

7/22/2016

Problem 4.7

value	Hash Key (value%7)
1	1
8	1
27	6
64	1
125	6
216	6
343	0

Table 1: Initial Table

0	None
1	None
2	None
3	None
4	None
5	None
6	None
7	None

Table 2: Add 1

0	None
1	1 – None
2	None
3	None
4	None
5	None
6	None
7	None

Table 3: Add 8

0	None
1	1 – 8 – None
2	None
3	None
4	None
5	None
6	None
7	None

Table 4: Add 27

0	None
1	1 – 8 – None
2	None
3	None
4	None
5	None
6	27 – None
7	None

Table 5: Add 64

0	None
1	1 – 8 – 64 – None
2	None
3	None
4	None
5	None
6	27 – None
7	None

Table 6: Add 125

0	None
1	1 – 8 – 64 – None
2	None
3	None
4	None
5	None
6	27 – 125 – None
7	None

Table 7: Add 216

0	None
1	1 – 8 – 64 – None
2	None
3	None
4	None
5	None
6	27 – 125 – 216 – None
7	None

Table 8: Add 343

0	343 – None
1	27 – 64 – None
2	None
3	None
4	None
5	None
6	1 – 8 – 125 – 216 – None
7	None

Problem 4.8

Value	Hash Key (value%5)
23	2
48	2
35	1
4	4
10	1

Table 9: Initial Table

0	None
1	None
2	None
3	None
4	None
5	None

Table 10: Add 23

0	None
1	None
2	None
3	23 – None
4	None
5	None

Table 11: Add 48

0	None
1	None
2	None
3	23 – 48 – None
4	None
5	None

Table 12: add 35

0	35 – None
1	None
2	None
3	23 – 48 – None
4	None
5	None

Table 13: Add 4

0	35 – None
1	None
2	None
3	23 – 48 – None
4	4 – None
5	None

Table 14: Add 10 (Final Table)

0	35 – 10 – None
1	None
2	None
3	23 – 48 – None
4	4 – None
5	None

Problem 4.1

A=1,2,3 B=3,4,5

a AuB = 1,2,3,4,5

b AnB = 3

c Difference(A,B) = 1,2,4,5

d Member(1,A) = true

e Insert(1,A) = 1,2,3

f Delete(1,A) = 2,3

g min(A) = 2

Problem 5

Following are several hash functions. None of them are very good as hash functions. Explain why they are not good hash functions and give an example showing them working poorly.

0.1 Part 1

Hash keys are character strings. The hash function h1(x) computes the length of a string.

This hash function would not be ideal for storing strings as every string with the same size will be given the same hash key. This also means as the hash key becomes larger, the amount of strings that can potentially be given that key increases at the rate of keys = 127ŝtrLen assuming that the string can be made up of all 127 ascii values.
My example below only considers characters a-z so there are 26ŝtrLen possible character combinations per key

Table 15: Problem 5-1

0	None
1	a – b – c – ... – z
2	aa – ab – ac – ... – az – ba – ... – zz
3	aaa – aab – aac – ... – zzz
4	None
5	None
6	None
7	None

0.2 Part 2

The function h2(x) computes a random number r with 1 r B, where B is the number of buckets. It returns r.
This is another example of a poor hash function. A Hash key is meant to be unique to a value, however, the hash function must return the same key for a value in order to pull the value from the hash map. But, because the value is randomly generated, there is no way of knowing whether the key returned by the hash function will be the same each time the function is called. This means that there is a good chance that when the new key is generated it may be linked to wrong value.
Ex:
key = GetHashKey(5) #= 1
Insert(5,key)

Table 16: Problem 5-2

0	None
1	5
2	None
3	None
4	None
5	None
6	None
7	None

key = GetHashKey(5) #= 5
getValue(5,key) # Returns None