Matt Bailey
EECS 560
Lab #3 Questions
9-20-2017

Q1) What is the algorithmic asymptotic complexity i.e. O(?) of each of the operations that you have implemented. (2 point)

putValue – O(n)

deleteValue – O(n)

print – O(n^2)

rehash - O(n^2)

a. The hash function (in terms of the number of characters in the string).

O(n)

b. Searching a value in the hash table (Assume hash function is ideal but its complexity is equal to the one you have stated in part a. and the bucket array size is much greater than the total number of elements)

O(n^2)

Q2) What are the properties of an ideal hash function in our use case (give at-least 4)? (Hint: Think on the lines of the performance for search operation. It degrades if there are too many collisions and the doubly link list that holds the data is too long. How can you keep the link list as short as possible?)(2 points)

An ideal hash function gives input values a near equal chance of landing anywhere in the hash table. The load factor will never exceed 5, the function should be quick to compute, and it should be hard to decrypt.

Q3) Describe your hash function. Why do you think it is ideal or what are the things it lacks with respect to the 4 properties you have mentioned above? (1 point)

My hash function is essentially the same as he one proposed by the instruction set. It is good, but imperfect. It doesn’t lead to perfect spreading on the table itself, and it is easy to decrypt.