OMP20007 Design of Algorithms
Semester 1 2016
Assignment 2
Ivan Ken Weng Chee 736901
Hashing Report

1. Insert : O(n) (If appending collision to linked list)
   O(1) (If prepending collision to linked list)
   Search : O(n)

2. Insert : O(n / size)
   Search : O(n / size)

3. Insert : O(1)
   Search : O(1)

4. Strings starting with the same character would hash to the same bucket.
   If storing dictionaries, the frequency distribution of words would not be uniform, thus causing more collisions on buckets.
   (Words starting with ‘s’ would collide much more than those starting with ‘q’)

5. Insert : O(n) (If appending collision to linked list)
   O(1) (If prepending collision to linked list)
   Search : O(n / size)

6. Algorithm simply applies a brute force approach by generates a string of random characters, of random length. Strings that hash to 0 are stored and checked for duplicates. Stops when n strings are obtained. Probability of obtaining a hash is Pr(1 / size).
   Expected running time : O(n / size)

7. Algorithm uses the Extended-Euclid algorithm to find the greatest common divisor of (r0, size) and checks if it is 1 (r0 and size are coprime). The hash is reversed and a one character string is generated. This is repeated until there are no more solutions left within the printable range. The algorithm then finds and checks if the greatest common divisor of (r1, size) is 1 (r1 and size are coprime). Then the solutions for two character strings within range is computed using the Euclid’s method of solving Linear Diophantine Equations.
   This algorithm only works for n <= 2. An error messaged will be printed to stderr when allocation fails.
   Expected complexity : O(n * (size) ^ 2)