Weekly Assignment 13:
Hashing

1. Demonstrate what happens when we insert the keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 into a hash table with collisions resolved by chaining. Let the table have 11 slots, and let the hash function be $h(k) = (2k + 5) \mod 11$.

2. (a) Draw the hash table of length $m = 13$ resulting from hashing the keys 8, 12, 40, 13, 88, 45, 29, 20, 23, and 77, using hash function $h(k) = (3k + 1) \bmod 13$ with open addressing and linear probing. Report for every key how often you had to probe unsuccessfully before you could insert the key, and report the total number of unsuccessful probes.

   (b) What is the result if instead we assume collisions are handled by double hashing with a primary hash function $h'(k) = h(k)$ and a secondary hash function $h''(k) = k \bmod 15$? Report for every key how often you had to probe unsuccessfully before you could insert the key, and report the total number of unsuccessful probes.

3. You have a universe of keys consisting of 30 numbers $\{0, 1, \ldots, 29\}$ and a hash table of size 10 that uses chaining. Which hash-function is better, $h_1(k) = k \bmod 10$, or $h_2(k) = \lfloor k/10 \rfloor$? Motivate your answer.

4. Suppose you are given a list A of n integers taken from a set U and you can use a hash table T of size n and a hash function $h : U \to \{0, \ldots, n-1\}$ that satisfies the assumption of simple uniform hashing. Design an $\mathcal{O}(n)$-time algorithm (average time) to decide whether list A contains two equal numbers. Explain your algorithm and discuss its complexity.

5. Design an algorithm that uses a hash table to compute, given a text file with n words, the 4 most frequently occurring words. You may assume that the INSERT and LOOKUP operations run in average time $\mathcal{O}(1)$. Your algorithm should run in average time $\mathcal{O}(n)$. Explain your algorithm and discuss its complexity.