CS350 – Winter 2019
Homework 3

Due Thursday, 28^{th} Feb 2019, on paper, at the start of class. This assignment will be graded.

1. Use the Master theorem to find the order of growth of the following recurrences:
   (a) $T(n) = 2T\left(\frac{2n}{3}\right) + 1, \quad T(1) = 1$
   (b) $T(n) = 2T\left(\frac{n}{2}\right) + n^2, \quad T(1) = 1$
   (c) $T(n) = 2T\left(\frac{n}{2}\right) + n, \quad T(1) = 1$
2. Explain *why* the answers are different. That is, what is so different about the *algorithms* that the above relations characterize?
3. A DNA sequence consists of four amino acids represented by the letters [A,C,G,T]. Each DNA sequence can be represented by a series of these letters. For example, the following pattern:

   TCCTATTCTT

   is a DNA sequence for a gene segment of chromosome 10.
   (a) Construct the shift table for the above gene segment.
   (b) Execute Horspool's Algorithm by hand to find the gene segment in the following DNA sequence.

   TTATAGATCTCGTATTCTTTTATAGATCTCCTATTCTT

   At each step, show (with a diagram) how the pattern aligns with the sequence being searched.
4. Consider the following U.S. state abbreviations as your **keys** to a hash table:

   AZ, AL, HI, NJ, OK, LA, NC, WA, RI, MA, WY, OR, KS

   (The entries accessed by these keys need not concern us; perhaps they are large records containing information about the corresponding state.) Use the following hash function:

   $$H(s) = s_1 * 3 + s_2$$

   where s_1 and s_2 are the positions of the first and second letters of the key s in the alphabet. (So, when $s =$ AZ, $s_1 = 1$ and $s_2 = 26$).
   (a) What is the range of the hash function H?
   (b) If the size of your hash table less than the size of the range, how do you convert the hash function into an index into the hash table?
   (c) Construct a hash table of size 23 using **open hashing** as your mechanism for handling collisions. Show all the steps, i.e., draw the hash table every time you add a new key.
   (d) What is the **load factor** of this hash table?
   (e) How many words of memory does it occupy (in addition to the space occupied by the entries). Assume that pointers occupy one word, and that your hash table stores pointers to the entries.

(f) Construct a hash table of size 23 using **closed hashing** as your mechanism for handling collisions. Show all the steps, i.e., draw the hash table every time you add a new key.

(g) What is the **load factor** of this hash table?

(h) How many words of memory does it occupy (in addition to the space occupied by the entries). Assume that pointers occupy one word, and that your hash table stores pointers to the entries.