Mcs-Cs 401, Summer 2016, Lowman, Quiz 2 (HW6).
due in class Monday, July 28.

Hash functions

- In section 11.3.2 of the textbook page 263-264 the multiplication method is given by:

$$h(k) = \lfloor m\ (kA\ mod\ 1) \rfloor \text{ with } 0 < A < 1. \tag{1}$$

- The CLRS slides for hash functions (used in lectures) describe the multiplication method for the special case where the table size is an integer power of 2. The hash function is given as:

  Assume all keys are integers, $m = 2^r$, and our computer has w bit words.

$$h(k) = (A \cdot k\ mod\ 2^w)\ rsh(w - r) \text{ with } 2^{w-1} < A < 2^w \tag{2}$$

1. Show that version (2) is the same as version (1) under the given conditions.
2. Are there any advantages to using the second version? Explain.
3. Donald Knuth, who has been called the "father of the analysis of algorithms", suggests that A $\approx \dfrac{(\sqrt{5}-1)}{2} = 0.6180339887\ldots$ is likely to work reasonably well in version (1).

   Do Exercise 11.3-4 assuming version (1).
4. Repeat Exercise 11.3-4 using a table size that is the next power of 2 greater than 1000.
5. Give a function definition for the previous problem that implements

$$h(k) = (A \cdot k\ mod\ 2^w)\ rsh(w - r) \text{ with } 2^{w-1} < A < 2^w$$

   The values of A,w, and r must be defined in your function definition.

   Note, the examples used in class notes the hash functions were coded in the C programming language. Many of the hash functions posted on the internet are also given as C code. C code is often used as pseudocode in textbooks. Define your function in C. Do not use Java. If you do not understand C code then use pseudocode similar to that used in the textbook.
6. In the previous problem k is an integer. Modify your function so the input is a key represented as a string. Your hash function should first convert the string to an int using Horner's Method and then hash the resulting integer to an array index.
7. Using the division method define a hash function that will only use the lower 8 bits of the key. All of the higher bits will be ignored resulting in a hash function that does not use all of the information available in the key. This is not a good hash function but easy to code.

8. **Heapsort** On page 155 of the textbook it states that

   (a) $\dots$ The chldren's subtrees each have size at most $\boldsymbol{\frac{2n}{3}}$

   (b) the worst case occurs when the bottom level of the tree is exactly half full and

   (c) therefore we can describe the running time of $\boldsymbol{Max - Heapify}$ by the recurrence
   $$\boldsymbol{T(n) \leq T\left(\frac{2n}{3}\right) + \Theta(1)}$$

   (d) Derive (a), show (b), (c) and finally solve (c).