Kevin Franklin

Studio: Ch 1

This studio consists of several practice exercises.

1. Given the un-reduced big-O value, calculate the reduced value.
   1. 10 – O(1)
   2. 10n - O(n)
   3. 5n2 + 2n – O(n2)
   4. 3 log2n + 2 – O(log2n)
2. For each of the following algorithms, calculate the big-O value. Be sure to specify which value *n* refers to.
   1. **Reversing a string –** O(n) where n is the number of char in the string
   2. **Printing out a matrix** – O(nx) where n is how many rows and x is how many dimensions the matrix is
   3. **Reversing each string in an array –** O(n) where n is number of strings
3. Suppose you have an array of Customer objects, sorted in alphabetical order by last name. For each of the following tasks, determine the run time in terms of big-O.
   1. Print the names of all of the customers. - O(n)
   2. Print the names of only the customers with last name starting with “A”. - O(n)
   3. Find all customers with last name beginning with “A”. – O(n)
   4. Find all customers with first name beginning with “A”. – O(n)
4. Now suppose that you have a dictionary (or hash map) of customer objects, where the keys are letters and the values are arrays storing all customers with last name beginning with that letter. For example, if our dictionary is customers then customers["A"] is an array of all customers with last name ending with “A”. Within each array, the customer objects are not sorted in any way.
   1. Print the names of all the customers. - O(n)
   2. Print the names of only the customers with last name starting with “A”. - O(n)
   3. Find all customers with last name beginning with “A”. - O(1)
   4. Find all customers with first name beginning with “A”. - O(n)