**1.5. Studio: Analyzing Algorithms**

This studio consists of several practice exercises.

1. Given the un-reduced big-O value, calculate the reduced value.
   1. 10

1

* 1. 10n

n

* 1. 5n2 + 2n

n2

* 1. 3 log2n + 2

logn

1. For each of the following algorithms, calculate the big-O value. Be sure to specify which value *n* refers to.
   1. **Reversing a string**

reversed = ''

for c in str:

reversed = c + reversed

n, where n is how many letters are in the string

* 1. **Printing out a matrix**

for row\_idx in matrix.length:

row = matrix[row\_idx]

for col\_idx in row:

print row[col\_idx]

print " "

print "\n"

n2, where n is the number of rows or columns, whichever is largest

in reality though, it should be (row \* col)

n2 only makes sense if the rows and columns are the same

* 1. **Reversing each string in an array**

*Note:* For this exercise, assume that each string has at most 10 characters.

reversed\_strings = []

for str in strings:

reversed = ''

for c in str:

reversed = c + reversed

reversed\_strings.append(reversed)

n, where n is the number of strings in the array

1. 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.

1

* 1. Print the names of only the customers with last name starting with “A”.

1

* 1. Find all customers with last name beginning with “A”.

1

* 1. Find all customers with first name beginning with “A”.

1

1. 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 of the customers.

n

* 1. Print the names of only the customers with last name starting with “A”.

1

* 1. Find all customers with last name beginning with “A”.

1

* 1. Find all customers with first name beginning with “A”.

n