CS 310: Extra Credit

Student Name:	

Extra Credit Overview

Students must complete these questions independently -- neither the instructor nor the teaching assistants shall render assistance. Moreover, students remain prohibited from discussing these problems with one another. Students may, however, use the textbook, lecture notes, and internet references to comprehend the subject matter.

Complexity Analysis

1. (5 pts): For each code fragment below, give the complexity of the algorithm (O or Θ). Give the tightest possible upper bound as the input size variable increases. The input size variable in these questions is exclusively ***n***.

Complexity	Code

	`public static int recursiveFunction(int n){`
	`  if( n <= 0 ) return 0;`
	`  return recursiveFunction(n - 1) + 1;`
	`}`

	`for(int i=0; i < n; i++)`
	`  for(int j=0; j < i; j++)`
	`    for(int k=0; k < j; k++)`
	`      for(int m=0; m < k; m++)`
	`        System.out.println( “Hello world” );`

	`public static int otherRecursiveFunction(int n){`
	`  if( n <= 0 ) return 0;`
	`  int sum = 0;`
	`  for(int i = 0; i < n; i++)`
	`    sum += otherRecursiveFunction(n - 1);`
	`  return sum;`
	`}`

2. (5 pts): Answer the following questions. The input size variable in these questions is exclusively ***n***.

Complexity	Operation
	Searching for the lowest priced vest in a collection of *n* vests.
	Finding the highest priced tie in a list *n* of ties sorted by price.
	Remove a single element from the rear of a singly linked list of *n* items. The linked list uses a head and tail pointer.
	Add a single element to the front of a singly linked list of *n* items. The linked list uses a head and tail pointer.
	Push two items onto a stack of *n* items.

6. (2 pts) Answer the following questions about the binary search algorithm using an ordered array with 8,675,309 items.

a) How many comparisons must one perform, in the worst case, to find any one item?

b) How many comparsions must one perform, in the best case, to find any one item?

7. (3 pts) A certain linear algorithm takes 3 seconds to process with an input size of n = 1500

a) How long would it take if n = 6000?

b) How long would it take with an input size of 1500 if the algorithm were quadratic instead of linear?

c) How long would it take with an input size of 1500 if the algorithm were cubic instead of linear?