COMP 3270 Introduction to Algorithms

Homework 1

1. (18 points) Understand the following algorithm. Simulate it mentally on the following four inputs, and state the outputs produced (value returned) in each case: (a) A: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]; (b) A: [-1, -2, -3, -4, -5, -6, -7, -8, -9, -10]; (c) A: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0]; (d) A: [-1, 2, -3, 4, -5, 6, 7, -8, 9, -10].

```
Algorithm Mystery (A:array[1..n] of integer)
   sum, max: integer
1  sum = 0
2  max = 0
3  for i = 1 to n
4     sum = 0
5        for j = i to n
6           sum = sum + A[j]
7           if sum > max then
8              max = sum
9  return max
```

Output when input is array (a) above:

Output when input is array (b) above:

Output when input is array (c) above:

Output when input is array (d) above:

What does the algorithm return when the input array contains all negative integers?

What does the algorithm return when the input array contains all non-negative integers?

2. (9 points) Fill out the following table w.r.t. the above algorithm *Mystery* with the input size *n*.

Step	Total # of times executed
1	
2	
3	
4	
5	
6	
7	
8	
9	

3. (10 points) Compare the following pairs of functions in terms of order of magnitude. In each case, say whether $f(n) = O(g(n)), f(n) = \Theta(g(n))$, and/or $f(n) = \Omega(g(n))$.

	$f(n)$	$g(n)$	
a.	$100n + log\, n$	$n + (log\, n)^2$	
b.	$log\, n$	$log(n^2)$	
c.	$\frac{n^2}{log\, n}$	$n(log\, n)^2$	
d.	$n^{\frac{1}{2}}$	$log\, n^5$	
e.	$n2^n$	3^n	

4. (23 points) $T(n) = 7T(n/8) + cn; T(1) = c$. Determine the polynomial T(n) for the recursive algorithm characterized by these two recurrence relations, using the Recursion Tree Method. Drawing the recursion tree may help but you do not have to show the tree in your answer; instead, fill the table below. You will need to use the following results, where a and b are constants and x<1:

$$a^{\log_b n} = n^{\log_b a}$$

$$\sum_{i=0}^{\infty} x^i = \frac{1}{1-x}$$

Level	**Level number**	**Total # of recursive executions at this level**	**Input size to each recursive execution**	**Work done by each recursive execution, excluding the recursive calls**	**Total work at this level**
Root	0				
1 level below	1				
2 levels below	2				
The level just above the base case level					
Base case level					

T(n) =

5. (10 points) Find a counterexample to the following claim:

$f(n) = \mathrm{O}(s(n))$ and $g(n) = \mathrm{O}(r(n))$ imply $f(n) - g(n) = \mathrm{O}(s(n) - r(n))$