Name: ____________________

CS343: Analysis of Algorithms, Sp 04
Exam 1

Score:				
	1.	(max 10)	5.	(max 10)
	2.	(max 17)	6.	(max 10)
	3.	(max 10)	7.	(max 10)
	4.	(max 10)	8.	(max 13)
			9.	(max 10)
	Total:		(max 100)	

This exam is closed book. Calculators are not allowed.

1. (10 pts) Use *induction* to prove $\sum_{i=0}^{n} x^i = \frac{1-x^{n+1}}{1-x}$. Be sure to show and give justification for all of your steps.

2. (17 pts total) Assume that you have a list A of sorted integers. You want to determine if a particular integer value is in the list. The fastest way to do this is to use recursive binary search where you start by comparing your value with the item in the middle of the list. If it matches, you are done. If your value is larger than this item, you recursively search the top part of the list. If your value is smaller than this item, you recursively search the bottom part of the list.

   (a) (10 pts) Implement recursive binary search by completing the `findVal` method below. Assume that the array `A` is in sorted order. The method should return `true` if `val` found in the array `A`. Otherwise, it returns false.

```
public boolean find(int val, int [] A) {
    return findval(val, A, 0, A.length-1);
}
public boolean findVal(int val, int[] A, int i, int k) {



}
```

   (b) (2 pts) Write down a recurrence relation (i.e. T(n) = ...) for the above algorithm where T(n) is the number of comparisons. Don't worry about the details, but instead try to capture the main behavior, much the way we did for MergeSort or StoogeSort. You should be able to do this even if you didn't complete part a) - just reason about what the algorithm is doing.

   (c) (5 pts) Use the Master Theorem to obtain the complexity of your algorithm. (You should be able to test your answer on a small problem).

3. (10 pts total) Suppose you have designed a new sorting algorithm and you want to empirically measure its complexity. So, you run the algorithm to calculate the average number of comparisons, $c(n)$, as a function of the input size n. Plotting your results, you find that the graph of n vs $\log_2 c$ is a straight line with a slope of $m = 2$ and y-intercept $b = 3$.

   (a) (8 pts) What can you say about the complexity of your algorithm in terms of $m = 2$ and $b = 3$? Be as specific as possible and explain your answer. Try to simplify your answer as best you can.

   (b) (2 pts) Which is better, your algorithm or StoogeSort? Explain.

4. (10 pts) Use telescoping to solve $T(n) = 3T(n-1) + 1$, where $T(1) = 1$;

5. (10 pts total)Use the Master Theorem to solve the following recurrences. Be sure to specify which case applies and show all of your work.

   (a) (5 pts) $T(n) = 8\,T(\frac{n}{2}) + n^2$

   (b) (5 pts) $T(n) = 2\,T(\frac{n}{4}) + n$

6. (10 pts) State the definition of $f(n) = o(g(n))$.

7. (10 pts) Use the definition you gave in the previous problem to prove that $2^n = o(2^{2n})$. Carefully explain *all* of your steps.

8. (13 pts) Indicate for each pair of expressions (A and B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geq 1$, and $c > 1$. You should fill in a "yes" or "no" in each slot of the table.

A	B	O	o	Ω	ω	Θ
$\log_{10} n$	$\lg n$					
$\lg^6 n$	$\sqrt{n}$					
n^k	c^n					
$c^{\lg n}$	$n^{\lg c}$					
$n!$	2^{2^n}					

9. (1 pts each, 10 pts total) Circle true or false:
   (a) true or false: $f(n) = \Omega(f(n)/2)$
   (b) true or false: $f(n) = o(g(n)) \Rightarrow g(n) = \Omega(f(n))$
   (c) true or false: $f(n) = O(g(n)) \Rightarrow f(n) = o(g(n))$
   (d) true or false: $f(n) = O(g(n)) \Rightarrow \lim_{n\to\infty} \frac{g(n)}{f(n)} = c$
   (e) true or false: $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0 \Rightarrow f(n) = o(g(n))$
   (f) true or false: $f(n) = o(f(n)^2)$
   (g) true or false: $f(n) = \Theta(g(n)) \Rightarrow f(n) = \Omega(g(n))$
   (h) true or false: $f(n) = \omega(g(n)) \Rightarrow f(n) = \Omega(g(n))$
   (i) true or false: My algorithm runs in $O(n)$ time and hers runs in $o(n^2)$. Therefore, hers *must* be worse.
   (j) true or false: My algorithm runs in $O(n)$ time and hers runs in $\omega(n)$. Therefore, hers *must* be worse.

10. (0 pts) What is the complexity of this exam?