# Notre Dame University
# CSC325 Analysis of algorithms
# Spring 2010 - Exam 1

Name:

**Multiple choice questions** (10pts)

1. Let $f(n) = n^2 + 2n + 3$ and $g(n) = n^{4/3} + \sqrt{n}$. We can assert that
   (a) $f(n) = \Theta(g(n))$
   (b) $f(n) = O(g(n))$
   (c) $f(n) = \Omega(g(n))$
   (d) None of the above.
2. Let $f(n) = n^n + n! + 2$ and $g(n) = 4 \times \sqrt{n}$. We can assert that
   (a) $f(n) \times g(n) = \Theta(n^n)$
   (b) $f(n) \times g(n) = \Theta(n! \times \sqrt{n})$
   (c) $f(n) \times g(n) = \Theta(n^n \times \sqrt{n})$
   (d) None of the above.
3. Let $f(n) = 2^{2n}$ and $g(n) = 2 \times 2^{n+1000}$. We can assert that
   (a) $f(n) + g(n) = \Theta(2^{2n})$
   (b) $f(n) + g(n) = O(2^{n+1000})$
   (c) $f(n) + g(n) = \Theta(2^n)$
   (d) None of the above.
4. Which of the following sorting algorithms uses additional constant space to sort a list of $n$ elements?
   (a) Counting sort.
   (b) Merge-sort.
   (c) Quick-sort.
   (d) None of the above.
5. Let the list of integers 22, 10, 34, 16, 5, 40, 3, 25. After 2 iterations of Merge-sort we get:
   (a) 3, 5, 10, 16, 22, 25, 34, 40
   (b) 10, 22, 16, 34, 5, 40, 3, 25

(c) 22, 10, 34, 16, 40, 5, 25, 3

(d) 10, 16, 22, 34, 3, 5, 25, 40

**Exercise 1** (20pts)
Arrange the following functions by increasing growth rate:

$$log\,(n!), log\,(n+10)^{1000}, 2^n + n^2, 0.1n^4 + 3n^3 + 1, log^2\,n, nlog\,n, 3^n$$

**Exercise 2** (10pts)
Solve the recurrences:

1. $T(n) = 3T(n/2) + O(n^2)$.

2. $T(n) = 9T(n/4) + O(n)$.

3. $T(n) = 4T(n/2) + O(n^2)$.

**Exercise 3** (20pts)
Sort the following list of integers using Quick-sort.

$$22, 10, 34, 16, 5, 40, 3, 25$$

At each recursive call, you are asked to indicate the pivot and the resulting list.

**Exercise 4** (10pts)
Prove by mathematical induction that $\sum_{i=1}^{n} i = \Theta(n^2)$.

**Exercise 5** (30pts)
Let $A[0..n-1]$ be an array of $n$ distinct real numbers. A pair $(A[i], A[j])$ is said to be an inversion if these numbers are out of order, i.e., $i < j$ but $A[i] > A[j]$.

1. Indicate the number of inversions in $[21, 14, 15, 18, 9, 5]$.

2. Design an algorithm for counting the number of inversions in an array. What is the running time of your algorithm?

3. Design an $O(n\ logn)$ algorithm for counting the number of inversions using a divide-and-conquer strategy.
   *Hint*: Modify mergesort to solve the problem.