[**Closed book, calculator, and notes**] Show all of your work clearly in the space provided or on the additional page at the end of the exam. If the additional page is used, clearly identify to which exam question it is related. Be sure to **read each problem carefully**. Note that the exam is double sided.

$$f(n) = \Theta(g(n)) \text{ and } g(n) = \Theta(h(n)) \quad \Rightarrow \quad f(n) = \Theta(h(n)) \tag{1}$$
$$f(n) = O(g(n)) \text{ and } g(n) = O(h(n)) \quad \Rightarrow \quad f(n) = O(h(n)) \tag{2}$$
$$f(n) = \Omega(g(n)) \text{ and } g(n) = \Omega(h(n)) \quad \Rightarrow \quad f(n) = \Omega(h(n)) \tag{3}$$

$$f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n)) \tag{4}$$

$$\lg n = \log_2 n \tag{5}$$
$$\ln n = \log_e n \tag{6}$$
$$a = b^{\log_b a} \tag{7}$$
$$\log_c(ab) = \log_c a + \log_c b \tag{8}$$
$$\log_b a^n = n \log_b a \tag{9}$$
$$\log_b a = \frac{\log_c a}{\log_c b} \tag{10}$$

$$\sum_{k=1}^{n} k = 1 + 2 + \cdots + n = \frac{n(n+1)}{2} = \Theta(n^2) \tag{11}$$

$$\sum_{k=0}^{n} x^k = 1 + x + \cdots + x^n = \frac{x^{n+1} - 1}{x - 1} = \Theta(x^n), \qquad x \neq 1 \tag{12}$$

$$\sum_{k=1}^{n} \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n} \approx \ln n + .577 = \Theta(\log n) \tag{13}$$

Given positive functions $f(n)$ and $g(n)$ such that

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = c$$

for some constant c.

1. If $0 < c < \infty$, then $f(n) = \Theta(g(n))$
2. If $0 \leq c < \infty$, then $f(n) = O(g(n))$
3. If $0 < c \leq \infty$, then $f(n) = \Omega(g(n))$

If $f(n)$ and $g(n)$ both approach zero or both approach ∞ in the limit, then

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \lim_{n \to \infty} \frac{f'(n)}{g'(n)}$$

where $f'(n)$ and $g'(n)$ denote derivatives of f and g with respect to n.

1. Although merge sort runs in $\Theta(n \lg n)$ worst-case time and insertion sort runs in $\Theta(n^2)$ worst-case time, the constant factors in insertion sort make it faster for small n. Thus, it makes sense to use insertion sort within merge sort when subproblems become sufficiently small. Consider a modification to merge sort in which n/k sublists of length k are sorted using insertion sort and then merged using the standard merging mechanism, where k is a value to be determined.

(a) (10 points) Here we have n/k sublists, each of length k. Give the worst case run time for insertion sort on **all** of the sublists (as a function of n and k). Be sure to justify your answer.

(b) (10 points) As a function of n and k, give the worst-case time complexity required to merge the n/k sublists back into one large list. Be sure to justify your answer.

2. (15 points) Let

$$p(n) = \sum_{i=0}^{d} a_i n^i$$

where $a_d > 0$, be a degree-d polynomial in n, and let k be a constant. Use the definitions of the asymptotic notation to prove that if $k \geq d$, then $p(n) = O(n^k)$.

3. (20 points) Find the asymptotic growth rate of the following recurrence using either iteration or substitution:

$$T(n) = \begin{cases} 1 & n = 1, \\ 3T(\lceil \frac{n}{3} \rceil) + c & \text{otherwise.} \end{cases} \tag{14}$$

where c is a real constant between 999 and 1002.

4. (10 points) Suppose that heapsort is modified so that a copy of an array is passed to it instead of passing a reference/pointer to the array. What will be the resulting worst-case time complexity for this modified version of heapsort? Justify your answer.

5. (10 points) When running Randomized-Quicksort on an array with n elements, explain why Random is called $\Theta(n)$ times in the worst case.

6.

(a) (15 points) Describe a $O(n \log n)$ (worst-case) alogrithm that takes an array of real numbers, A[1... n], and determines whether or not there exist two elements, A[i] and A[j], such that A[i] = -A[j]. (If A contains at least one zero, the answer is YES.) Your description should be in Java, pseudocode, or unambiguous English prose.
Note: A $O(n \log n)$ algorithm is required for full credit; however, a slower algorithm will be accepted for partial credit.

(b) (10 points) Clearly explain the asymptotic time complexity of your answer to part **(a)**.

Additional work area for any problem. Clearly identify to which problem the work on this page is related.

Additional work area for any problem. Clearly identify to which problem the work on this page is related.