[**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. (15 points) Find $\sum_{k=4}^{300}(2k+2)$. Be sure to note the bounds on the summation and show all your work. *Remember: you may not use a calculator.*

2. (20 points) Find the asymptotic growth rate of the following recurrence:

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

where c is a real constant between 2 and 999.

3. (10 points) Is the following sequence a max-heap? Justify your answer.
[32 26 8 9 25 12]

4. (20 points) At the end of the quarter, you need to store your favorite sorted array x of N words. Fortunately, Sand 'n Stuff, a company that manufactures storage devices, has a good deal on a device for storing data. You buy it, store x on it, and put in storage for the summer. Suppose that you return in the fall to find that the data has been scrambled.

You contact Sand 'n Stuff to complain. They assure you that all of the data is still there, it has just been rearranged slightly. You learn that a flaw in the memory device causes elements in your array to "float." For example, an item that was located at $x[1]$ can float to $x[3]$, forcing the item that used to sit in $x[3]$ to float somewhere also.

The good news is that the memory cells didn't float very far. You have be told that the maximum float distance is k ($k > 0$). That is, the final resting place of an element at the end of the summer is no more than k positions away from its location at the beginning of the summer. (I.e., the data that used to be at $x[i]$ is now somewhere between $x[i - k]$ and $x[i + k]$.)

What is the worst case time complexity (Θ) for InsertionSort when applied to your slightly scramble array, x? Your answer should be given as a function of the array size, N, and float distance, k. Be sure to justify your answer.

5. (15 points) During the running of the procedure QUICKSORT on an array of size n (assume n is a power of 2), how many calls are made to PARTITION in the best case. Note: This is slightly different from homework problem 7.3-2 since the value of n is constrained, it is asking about PARTITION, not RANDOM, and you are to specify your answer exactly, not in Θ notation.

6. Counting each arithmetic calculation or comparison and any pick-up or set-down of a card as one operation, what is the worse-case order of growth for the number of operations (using Θ notation) for an algorithm that sorts numbered cards in the following way?

(a) (10 points) Find the largest valued card in the deck by going through one card at a time extracting a card if it is the largest one seen so far and swapping the previously largest card back into the deck. When the largest has been found, place this card face down in a new pile and repeat the previous process until no cards in the original pile are left. Explain your answer.

(b) (10 points) This time we assume that the largest number on any of the n cards is n^2. We sort the cards by placing a set of n^2 plates numbered from 1 to n^2 on a table. Then one by one, place each card on top of the number equal to it on the desk. The sorted list can be extracted by looking through all n^2 plates in order. Treat each pick-up or set-down of a plate as one operation.

(bonus) (10 points) The method in part **(b)** can be improved to work in $\Theta(n)$ time. Explain how… this is not easy… hints: use division, try to turn each number into a pair of numbers with values between 1 and n.

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