[**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.

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

$$f(n) = \Theta(g(n)) \iff g(n) = \Theta(f(n)) \quad (4)$$

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

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

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

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

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. (10 points) Briefly, in your own words, explain the characteristics of a **stable** sorting algorithm.

2. (10 points) Consider the following algorithm:

```
SelectionSort(A[1..n]) {
  for i ← 1 to length[A]−1 do {
    min ← i
    for j ← i+1 to length[A] do {
      if( A[j] < A[min]) {
        min ← j
      }
    }
    swap A[i] with A[min]
  }
}
```

Give an example input array, A[1..n], for which the SelectionSort algorithm is unstable. (Full credit will be given for the smallest possible array size.) Be sure to explain your example.

3. (10 points) What is the maximum number of times during the execution of quicksort that the largest element can be moved, for an array of N elements? Be sure to justify your answer.

4. (20 points) Find the asymptotic growth rate of the following recurrence using either iteration:

$$T(n) = \begin{cases} 1 & n = 1, \\ T(\lceil \frac{n}{2} \rceil) + 1 & \text{otherwise.} \end{cases} \tag{14}$$

5. The problem is similar to what we did with Quicksort in order to produce Select in lecture.

(a) (10 points) Describe how Heapsort (using a min-heap) could be modified to implement Heap-Select(A[1..n], i) that returns the i^{th} order statistic in the array A.

(b) (10 points) What is the worst-case time complexity for finding the smallest value in A? Justify your answer.

(c) (10 points) What is the worst-case time complexity for finding the largest value in A? Justify your answer.

4. (20 points) At the end of last quarter, you needed to store your favorite sorted array A of n objects.[1] Fortunately, Sand 'n Stuff, a company that manufactures storage devices, had a good deal on a large capacity device for storing data. Suppose you bought it, and stored A on it at the end of the last academic year. Last night, when you should have been studying for this exam, you decided to look at the data stored on the device for the first time. You discovered that the data had been scrambled.

When you contacted Sand 'n Stuff the first thing this morning to complain, they assured 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 $A[1]$ can float to $A[3]$, forcing the item that used to sit in $A[3]$ 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 ($k \geq 0$). 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 $A[i]$ is now somewhere between $A[i-k]$ and $A[i+k]$.)

You realize that you can just sort A in order to restore your data. The collection of data is huge, i.e., n is extremely large. You must choose which sorting algorithm is best to use here. Consider the following: InsertionSort, HeapSort, QuickSort, MergeSort, and Randomized-QuickSort. Discuss the pros and cons of each and select your favorite for this task. You should justify your answer by appealing to time-complexity analysis and specific information you know about characteristics of the data being sorted.

[1]For privacy reasons, I was not able to determine what the objects were, but I have been assured that they can be sorted.

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.