[**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 $\infty$ 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) Find a simple formula for $\sum_{k=1}^{n}(2k-1)$.

**2.** (10 points) Consider the Counting-Sort algorithm described in lecture:

**Counting-Sort(A, B, k)**

```
for i ← 1 to k+1
  do C[i] ← 0
for j ← 1 to length[A]
  do C[A[j]+1] ← C[A[j]+1] + 1
// C[i+1] contains the number of elements equal to i
for i ← 1 to k
  do C[i+1] ← C[i+1] + C[i]
// C[i] contains the number of elements less than or equal to i
for j ← length[A] downto 1
    do B[C[A[j]+1]] ← A[j]
       C[A[j]+1] ← C[A[j]+1] − 1
```

The pseudocode provided does a stable sort. Write (pseudocode, Java, or C#) a simplified version of Counting-Sort that does not perform a stable sort.

**3.** (15 points) Write an algorithm (in English, pseudocode, Java, or C#) that will sort an array of 32-bit signed integers in $O(n)$ time, where $n$ is the size of the array. You must make use of the stable Counting-Sort algorithm from the previous question, however, the size of the C array is limited to 65536 elements.

Recall:

| Base 10 | Twos Complement |
|---|---|
| -128 | 10000000 |
| -127 | 10000001 |
| ⋮ | ⋮ |
| -1 | 11111111 |
| 0 | 00000000 |
| 1 | 00000001 |
| ⋮ | ⋮ |
| 127 | 01111111 |

**4.** Find the asymptotic growth rate of the following recurrence:

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

**(a)** (5 points) using the Master method

**(b)** (20 points) using iteration

**5.** This 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.

**6.** (20 points) Late last year, Google, Inc. announced that they had sorted 10,000,000,000,000 100-byte records in just over six hours. They used 4,000 computers and 48,000 hard drives to store and sort the data. [1]

**(a)** (5 points) Consider the following sorting algorithms: QuickSort, MergeSort, and InsertionSort. Which of these three is best suited for Google's sorting problem and why?

**(b)** (15 points) Answer the following (in English) as if it were an interview question: Given the constraints listed above, describe how you would make use of known sorting algorithms to sort the 10 trillion records?

Grading for (b) — Definite hire: 15, Second interview: 12, Consider next year: 10, Maybe: 7, Never: 3

[1] http://googleblog.blogspot.com/2008/11/sorting-1pb-with-mapreduce.html

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.