CSOR W4231.002 – Spring, 2016

Homework 1

Out: Monday, January 25, 2015
Due: 8pm, Monday, February 8, 2015

Please keep your answers clear and concise. *Collaboration is limited to discussion of ideas only. You should write up the solutions entirely on your own. You should list your collaborators on your write-up. If you do not type your solutions, be sure that your hand-writing is legible, your scan is high-quality (if applicable) and your name is clearly written on your homework.*

1. (25 points) In the table below, indicate the relationship between functions f and g for each pair (f, g) by writing "yes" or "no" in each box. For example, if $f = O(g)$ then write "yes" in the first box. Here $\log^b x = (\log_2 x)^b$.

f	g	O	o	Ω	ω	Θ
$\log^2 n$	$25 \log n$					
$\sqrt{\log n}$	$(\log \log n)^4$					
$3n \log n$	$n \log 3n$					
$n^{3/5}$	$\sqrt{n} \log n$					
$\sqrt{n} + \log n$	$2\sqrt{n}$					
$n^2 2^n$	3^n					
$\sqrt{n} 2^n$	$2^{n/2+\log n}$					
$n \log 3n$	$\frac{n^2}{\log n}$					
$n!$	2^n					
$\log n!$	$\log n^n$					

2. (12 points) Show that, if λ is a positive real number, then $f(n) = 1 + \lambda + \lambda^2 + \ldots + \lambda^n$ is

   (a) $\Theta(1)$ if $\lambda < 1$.

   (b) $\Theta(n)$ if $\lambda = 1$.

   (c) $\Theta(\lambda^n)$ if $\lambda > 1$.

   Therefore, in big-Θ notation, the sum of a geometric series is simply the first term if the series is strictly decreasing, the last term if the series is strictly increasing, or the number of terms if the series is unchanging.

3. (12 points)

   - Find (with proof) a function f_1 such that $f_1(2n)$ is $O(f_1(n))$.
   - Find (with proof) a function f_2 such that $f_2(2n)$ is not $O(f_2(n))$.
   - Give a proof or a counterexample: if f is $o(g)$ then f is $O(g)$.

4. (24 points) Consider an array A consisting of n integers, $A[1], A[2], \ldots, A[n]$. You want to output a two-dimensional n-by-n array B such that for all $i < j$, $B[i, j]$ contains the sum of array entries $A[i]$ through $A[j]$, that is, $A[i] + A[i + 1] + \ldots + A[j]$. (For $i \geq j$, the value of entry $B[i, j]$ is left unspecified, so we don't care what is output for these values.)

   The most natural way to solve this problem is provided by the following algorithm.

```
for i = 1, 2, ..., n do
    for j = i + 1, i + 2, ..., n do
        Add up array entries A[i] through A[j]
        Store the result in B[i, j]
    end for
end for
```

   (a) (6 points) Let $T(n)$ denote the running time of this algorithm on an input of size n. For a function f of your choice, show that $T(n)$ is $O(f(n))$.

   (b) (6 points) For the same function f, show that $T(n)$ is also $\Omega(f(n))$. Hence $\Theta(f(n))$ is an asymptotically tight bound for $T(n)$.

   (c) (12 points) Give a different algorithm for this problem with running time $T'(n)$ that is asymptotically better than $T(n)$. In other words, you should show that the running time of your new algorithm satisfies $T'(n) = O(g(n))$ for some function $g(n) = o(f(n))$.
   Hint: The new algorithm should exploit the structure of the problem better.

5. (16 points) Give tight asymptotic bounds for the following recurrences.

   - $T(n) = 4T(n/2) + n^3 - 1$.
   - $T(n) = 8T(n/2) + n^2$.
   - $T(n) = 6T(n/3) + n$.
   - $T(n) = T(\sqrt{n}) + 1$.

6. (31 points) The Fibonacci numbers are defined by the recurrence

$$F_0 = 0, F_1 = 1$$
$$F_n = F_{n-1} + F_{n-2} (n \geq 2)$$

   (a) (4 points) Show that $F_n \geq 2^{n/2}$, $n \geq 6$.

   (b) Assume that the cost of adding, subtracting, or multiplying two integers is $O(1)$, independent of the size of the integers.

   - (4 points) Write pseudocode for an algorithm that computes F_n based on the recursive definition above. Develop a recurrence for the running time of your algorithm and give an asymptotic lower bound for it.
   - (4 points) Write pseudocode for a non-recursive algorithm that asymptotically performs fewer additions than the recursive algorithm. Discuss the running time of the new algorithm.
   - (10 points) Show how to compute F_n in $O(\log n)$ time using only integer additions and multiplications.
     (Hint: Express F_n in matrix notation and consider the matrix

$$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}$$

     and its powers.)

   (c) (9 points) Now assume that adding two m-bit integers requires $\Theta(m)$ time and that multiplying two m-bit integers requires $\Theta(m^2)$ time. What is the running time of the three algorithms under this more reasonable cost measure for the elementary arithmetic operations?

7. **(Optional exercise: do NOT return, it will not be graded.)** Solve the following recurrences exactly, and prove that your solutions are correct by induction.

   - $T(1) = 2, T(n) = T(n-1) + 2n$.
   - $T(1) = 1, T(n) = 2T(n-1) + 1$.

8. **(Optional exercise: do NOT return, it will not be graded.)** Unlike a decreasing geometric series, the sum of the *harmonic series* $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \ldots$ diverges; that is, $\sum_{i=1}^{\infty} \frac{1}{i} = \infty$. It turns out that, for large n, the sum of the first n terms of this series can be well approximated as $\sum_{i=1}^{n} \frac{1}{i} \approx \ln n + \gamma$ where ln is the natural logarithm (log base $e = 2.718\ldots$) and γ is a particular constant $0.57721\ldots$.

   Show that

$$\sum_{i=1}^{n} \frac{1}{i} = \Theta(\log n).$$

   (Hint: To show an upper bound, decrease each denominator to the next power of two. For a lower bound, increase each denominator to the next power of two.)