University of Waterloo
CS 341 — Algorithms
Fall 2013
Assignment 1

Distributed: Wednesday, September 18, 2013.

Due: Wednesday, September 25, 2013, at 3:00pm.

1 Written assignment

1. Order Notation

   (a) Suppose $f(n) = \Theta(g(n))$. Prove formally that $f(n) + n \log n = \Theta(g(n) + n \log n)$.

   (b) State whether or not $f(n)$ is $O(g(n))$, $f(n)$ is $\Omega(g(n))$, $f(n)$ is $\Theta(g(n))$, $f(n)$ is $o(g(n))$, and $f(n)$ is $\omega(g(n))$. For each case above give a yes/no answer and a brief justification. $f(n) = \frac{n^4 \log n}{n+2}$ and $g(n) = \frac{n^4 - 2n^3}{5n^2}$.

   (c) State whether or not $f(n)$ is $O(g(n))$, $f(n)$ is $\Omega(g(n))$, $f(n)$ is $\Theta(g(n))$, $f(n)$ is $o(g(n))$, and $f(n)$ is $\omega(g(n))$. For each case above give a yes/no answer and a brief justification. $f(n) = n^{3-1/\log n}$ and $g(n) = n^3$.

   (d)

2. Suppose $f_1(n) \geq 0$ is $\Theta(g(n))$ and $f_2(n) \geq 0$ is $\Theta(g(n))$. Prove or give a counterexample:

   (a) $f_1(n) + f_2(n)$ is $\Theta(g(n))$

   (b) $f_1(n) - f_2(n)$ is $O(1)$

   (c) $f_1(n)/f_2(n)$ is $\Theta(1)$

3. Analyze the following pieces of pseudocode and for each of them give a tight (Θ) bound on the running time as a function of n. Show you work.

   Note: `lg`(n) is defined as the discrete logarithm base 2, i.e. the floor of $\log_2(n) + 1$.

   (a)
   ```
   for i = 1 to n
     for j = 1 to lg(n)
       for k = 1 to 19
         x = x + 1
   ```

(b)
```
for i = 1 to n
  for j = 1 to i^3
    for k = 1 to n
      x = x + 1
```

(c)
```
i = n
while(i > 1)
  for j = 1 to n
    x = x + 1
    if i is odd
      i = i - 1
    else
      i = i/2
```

(d)
```
i = n
while(i > 2)
  for j = 1 to n
    x = x + 1
  i = sqrt(i)
```

4. You are asked to compute the number b^n for n a nonnegative integer. The straightforward solution is to compute the iterated product $b \times b \times \ldots \times b$ from left to right, which takes time $\Theta(n)$. Give a divide-and-conquer algorithm for the same problem that takes time $\Theta(\log n)$.