**COMP 220** **Data Structures**

# Lab/Hwk 1 KEY

*Assigned: August 23* *Due: August 30*

**Abstract**

For lab you'll review some basics of complexity and algorithm analysis as well as hone your skills with Logarithmic thinking. For homework you'll dust off your C++ skills and explore the new programming environment: *Code::Blocks*.

## 1 Lab

1. The formula for the conversion from one logarithm base to another tells us that for any two bases $a$ and $b$, $\log_b(n)$ differs from $\log_a(n)$ by exactly a factor of $\frac{1}{\log_a(b)}$.

$$\log_b(n) = \frac{\log_a(n)}{\log_a(b)}$$

We're often interested in logs base 2, so let's compute the conversion factor for base $b = 2$ for each bases listed in the table below:

| Base $a$ | Factor to convert to $\log_2$ |
|---|---|
| e | $1/\log_e 2 \approx 1.44$ |
| 8 | $1/\log_8 2 = \log_2 8 = 3$ |
| 10 | $1/\log_1 02 \approx 3.32$ |
| 16 | $1/\log_1 62 = 4$ |
| 23 | $1/\log_2 32 \approx 4.52$ |
| 127 | $1/\log_1 272 \approx 6.99$ |
| 256 | $1/\log_2 562 = 8$ |

What do you notice about converting to $\log_2$ from some base $a = 2^k$?

The conversion factor is $1/\log_{2^k} 2 = \log_2 2^k = k$.

2. Fill in the following table in three phases. First without using a calculator, fill in the integer lower and upper bound. If the logarithm is an exact integer, then put that as both the lower and upper bound. Then, take a shot a guessing the logarithm's actual value out to 2 decimal places. Finally, use a calculator to compute the actual log to 2 places and fill that in for the final column. When computing actual logarithms, you may only use the log base 10 function on a calculator; use the conversion formula for all other bases.

| Logarithm | Lower Int. | Upper Int. | Guesstimate | Actual Value |
|---|---|---|---|---|
| $\log_2(17)$ | 4 | 5 | 4.01 | 4.09 |
| $\log_2(725)$ | 9 | 10 | 9.45 | 9.50 |
| $\log_2(1024)$ | 10 | 10 | 10 | 10 |
| $\log_2(7)$ | 2 | 3 | 2.75 | 2.81 |
| $\log_8(7)$ | 0 | 1 | 0.875 | 0.94 |
| $\log_8(178)$ | 2 | 3 | 2.25 | 2.50 |
| $\log_{16}(25)$ | 1 | 2 | 1.05 | 1.16 |
| $\log_{16}(333)$ | 2 | 3 | 2.65 | 2.09 |
| $\log_{10}(15)$ | 1 | 2 | 1.07 | 1.18 |
| $\log_{10}(150)$ | 2 | 3 | 2.07 | 2.18 |
| $\log_{10}(145787)$ | 5 | 6 | 5.12 | 5.16 |

3. Rank the following complexity from least to greatest order, in terms of resources needs by members of the class, where 1 is the least resource intense class and numbers proceed up from there.

| | |
|---|---|
| $O(n \log n)$ | 4 |
| $O(n^2)$ | 5 |
| $O(1)$ | 1 |
| $O(2^n)$ | 7 |
| $O(n^3)$ | 6 |
| $O(\log n)$ | 2 |
| $O(n)$ | 3 |

4. Determine the Big-O for each of the following functions.

| | |
|---|---|
| $\frac{7}{100}n + 1.5E16$ | $O(n)$ |
| $\frac{x \log x}{25} - 13x + 5x^3$ | $O(x^3)$ |
| $\frac{(n+1)(n+2)}{n}$ | $O(n)$ |
| $10^9$ | $O(1)$ |
| $\log y + 135 + \frac{y}{2}$ | $O(y)$ |

5. Consider the following loop template containing integer constants $a$ and $s$:

```
for(int i{a} ; i < n ; i+=s){
  // do some work, O(1)
}
```

(a) What constraints must be placed on the values of $a$, $s$, and $n$ in order for this loop to execute at least one iteration and terminate properly?

$a < n$ and $s > 0$

(b) Assuming the above constraints are met, exactly how many iterations will it perform before it terminates?

After $k$ iterations the loop terminates. The termination condition is that $i \geq n$, and we start with $i = a$, so we have

$$\begin{aligned} a + sk &\geq n \\ sk &\geq n - a \\ k &\geq \frac{n-a}{s} \\ \implies k &= \left\lceil \frac{n-a}{s} \right\rceil \end{aligned}$$

(c) What is the complexity of this loop and what effect, if any, do the values of $a$ and $s$ have on its complexity?

As long as $a$ and $s$ are constants (have no relation to the input size $n$), they have no effect. The complexity is $O(n)$.

6. Consider the following loop template containing integer constants $a$ and $s$:

```
for(int i{n} ; i > a ; i/=s){
  // do some work, O(1)
}
```

(a) What constraints must be placed on the values of $a$, $s$, and $n$ in order for this loop to execute at least one iteration and terminate properly?

$n > a$ and $s > 1$

(b) Assuming the above constraints are met, exactly how many iterations will it perform before it terminates?

After $k$ iterations the loop terminates. The termination condition is that $i \leq a$, and we start with $i = n$, so we have

$$\begin{aligned} \frac{n}{s^k} &\leq a \\ s^k &\geq \frac{n}{a} \\ k &\geq \log_s \frac{n}{a} \\ \implies k &= \left\lceil \log_s \frac{n}{a} \right\rceil \end{aligned}$$

(c) What is the complexity of this loop and what effect, if any, do the values of $a$ and $s$ have on its complexity?

$O(\log n)$. $a$ and $s$ have no effect since they're constants.

## 2 Homework 1

The goal of this homework is to get back into C++ and to familiarize yourself with the use of the new IDE, Code::Blocks. To do this you should complete *exercise 7 from chapter 2 of the text* with a few modifications:

1. Be sure to put your function in a library.
2. Write tests for your function using the gTest testing framework.
3. Write a main program that provides a basic CLI interface to your function. If your program is called *sqrt* then running *sqrt 4* at the CLI should produce *2*. You may assume valid command line arguments for this assignment; Error checking and validation of command line arguments is not required but you are strongly encouraged to do it anyway.

Submit your *source code only* via the *handin* program. The course is, of course, *comp220* and the assignment is *hwk1*. The homework is due prior to next week's lab, so **8/30 by 2pm**.

```
double sqrt(double x) {
  if (x < 0) {
    std::cerr << "Can't take sqrt of negative number!" << x << std::endl;
    std::exit(1);
  }
  if (x == 0) return 0.0;

  double g1 = x / 2;
  double g2 = x / g1;
  double avg = (g1 + g2) / 2;

  while (avg != g1 && avg != g2) {
    g1 = avg;
    g2 = x / g1;
    avg = (g1 + g2) / 2;
  }
  return avg;
}
```