

University of Nevada, Las Vegas Computer Science 477/677 Fall 2024

Assignment 1: Due Saturday August 29, 2026 midnight

Follow our TA, Shareem Arjyal's (arjyal@unlv.nevada.edu) instructions on how to turn in the assignment.

Name: _____

You are permitted to work in groups, get help from others, read books, and use the internet.

1. The sequence of powers of 2 is generated by the recurrence $2^n = 2 \cdot 2^{n-1}$. What is the recurrence which generates the Fibonacci sequence $F_1, F_2, \dots$?

$$F_n = F_{n-2} + F_{n-1}$$

2. Write the sequence of all Fibonacci numbers under 100.

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89

3. Which one of these statements is true?

- (a) The Fibonacci numbers increase logarithmically.
- (b) The Fibonacci numbers increase linearly.
- (c) The Fibonacci numbers increase quadratically.
- (d) The Fibonacci numbers increase exponentially.

Answer: (d)

4. Find the constant K such that $F_n = \Theta(K^n)$. What is the standard name of this constant? One way to approach this problem is to assume that $F_n = K^n$ for all n . That's false, but it's close enough that we can get the right answer. We then have:

$$F_n = F_{n-2} + F_{n-1}$$

$$K^n = K^{n-2} + K^{n-1} \quad \text{Divide both sides by } K^{n-2} :$$

$$K^2 = 1 + K$$

$$K^2 - K - 1 = 0 \quad \text{Now use the quadratic formula}$$

$$K = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad \text{where } a = 1, b = -1, c = -1$$

$$K = \frac{-1 \pm \sqrt{1+4}}{2} \quad \text{But, since } K \text{ cannot be negative:}$$

$$K = \frac{-1 + \sqrt{5}}{2} = 1.616035 \dots$$

the so-called "golden ratio."

In this course, I expect you to understand logarithms. For numbers $x > 0$ and $b > 1$, the base b logarithm of x is written $\log_b x$. The base is usually not written, but is understood to be the “default” base, which depends on the application. For any base, $\log(xy) = (\log x)(\log y)$, $\log(x^y) = y \log x$, and $\log_b x = \log x / \log b$.

- (i) In science and engineering, the default base is 10. That is if $\log x$ appears in a scientific discussion or manuscript, it means $\log_{10} x$.
- (ii) In mathematics, the default base is $e \approx 2.718$, and $\log_e x$ is usually written $\ln x$.
- (iii) In computer science, or computing in general, the default base is 2. Thus $\log 2 = 1$, $\log 4 = 2$, $\log 8 = 3$, and $\log 65536 = 16$. If I write $\log x$ in a homework assignment or an exam, I mean $\log_2 x$.

5. If you write `log(x)` in a C++ program, what is the base of the logarithm?

In C++, at least in my version, the value of `log(x)` is the natural logarithm $\ln x$, whose base is e .

6. There are quantities that are normally expressed using a logarithmic scale. Describe four of them. Hint: Chemistry, seismology, acoustics, astronomy.

- (a) pH is defined to be the the logarithm of the reciprocal of the hydrogen ion concentration. In plain water, $[H^+] = 10^{-7}$, hence the pH of water is 7.0.

On the Richter scale, magnitude of an earthquake is proportional to the logarithm (base 10, of course) of its energy. Thus, a magnitude 7.0 earthquake is 10 times as powerful as a magnitude 6.0 earthquake.

- (b) The intensity of a sound is proportional to the logarithm, base 10, of the number of bels. A decibel is one tenth of a bel. Thus, a 70 decibel sound is 10 times as intense as a 60 decibel sound.
- (c) Brightness of a star is expressed logarithmically, but the base is $\sqrt[5]{100}$. The visual magnitude of a star is proportional to the negative of the logarithm of its apparent brightness. Thus, a first magnitude star appears 100 times as bright as a sixth magnitude star. Very bright stars and planets have negative magnitude. For example, Sirius, the brightest star in the sky, appears more than 2 million times as bright as the planet Pluto.

7. You’ve seen Landau notation. Originally, there was only “big O,” but now there are several others. We will only use three of those this semester.

When we write $f(n) = O(g(n))$, we mean that there are constants C and N such that $f(n) \leq C g(n)$ for all $n \geq N$.

When we write $f(n) = \Omega(n)$, we mean that there are constants C and N such that $f(n) \geq C g(n)$ for $n \geq N$.

When we write $f(n) = \Theta(g(n))$ we mean that both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

For example, $n^2 + 5n + 3$ is both $O(n^2)$, and $\Omega(n^2)$, and thus $n^2 + 5n + 3 = \Theta(n^2)$.

8. Write either O , Ω or Θ in each blank. Write Θ if that is correct, otherwise write O or Ω . Think carefully. Some of these require thought.

- (a) $n - 100 = \Theta(n - 200)$

(b) $n^{1/2} = \Omega(n^{2/3})$

(c) $100n + \log n = \Theta(n + \log^2 n)$

Hint: $\log n$ grows more slowly than any polynomially increasing function of n .

(d) $n \log n = \Theta(10n + \log(10n))$

(e) $\log(n^2) = \Theta(\log(n^3))$

(f) $10 \log n = O(\log(n^2))$

(g) $n^{1.01} = \Omega(n \log^2 n)$

(h) $n^2 / \log n = O(n \log^2 n)$

This one requires serious thinking. Don't just write down the first thing that occurs to you.

(i) $n^{0.1} = \Omega(\log^2 n)$

The rest of these are harder, and may require calculation.

(j) $(\log n)^{\log n} = \text{-----} (n / \log n)$

(k) $\sqrt{n} = \Omega, (\log^3 n)$

This one requires writing things down.

(l) $n^{1/2} = O(5^{\log_2 n})$

Think!

(m) $n2^n = O(3^n)$

This one is (slightly) tricky.

(n) $2^n = \Theta(2^{n+1})$

This one is easy, if you think about it correctly.

(o) $n! = \Omega(2^n)$

Don't forget that the default base is 2.

(p) $\log n^{\log_2 n} = \Theta(2^{(\log n)^2})$

In fact, $\log n^{\log_2 n}$ is exactly equal to $2^{(\log n)^2}$.

If you know your calculus, this one is easy.

(q) $\sum_{i=1}^n i^k = \Theta(n^{k+1})$

The summation approximates an integral. From calculus, you know that $\int_0^n x^k dx = \frac{x^{k+1}}{k+1}$. The denominator is a constant, hence has no effect on the asymptotic value.

This next one is quite important for analyzing the time complexity of sorting algorithms, and it **will** appear on exams, and very likely during job interviews.

$$(r) \log n! = \Theta(n \log n)$$

9. Consider the following C++ program.

```
void process(int n)
{
    if(n > 1) process(n/2);
    cout << n%2;
}

int main()
{
    int n;
    cout << "Enter a positive integer: ";
    cin >> n;
    assert(n > 0);
    process(n);
    cout << endl;
    return 1;
}
```

The last line of the output of `process(n)` is a string of bits. What does this bitstring represent?

`process(n)` writes the binary numeral for n .

10. The C++ code below implements a function, “mystery.” What does it compute?

```
float mystery(float x, int k)
{
    if (k == 0) return 1.0;
    else if(x == 0.0) return 0.0;
    else if (k < 0) return 1/mystery(x,-k);
    else if (k%2) return x*mystery(x,k-1);
    else return mystery(x*x,k/2);
}
```

`mystery(x,k)` returns the value of x^n