

# University of Nevada, Las Vegas Computer Science 477/677 Fall 2021

## Assignment 1: Due Monday August 30 2021

Name: \_\_\_\_\_

You are permitted to work in groups, get help from others, read books, and use the internet. Your answers must be written in a pdf file and uploaded to canvas, by midnight January 26th. Your file must not exceed 4 pages. If you have any questions, or you are having trouble uploading the assignment you may email the grader, XXXXX at XXXXX@unlv.nevada.edu.

1. Problem 0.1 on page 8 of the textbook. In each of the following situations, write  $O$ ,  $\Omega$ ,  $\Theta$  in the blank.

(a)  $n - 100 = \text{-----} (n - 200)$

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

(c)  $100n + \log n = \text{-----} (n + \log^2 n)$

(d)  $n \log n = \text{-----} (10n + \log(10n))$

(e)  $\log(2n) = \text{-----} (\log(3n))$

(f)  $10 \log n = \text{-----} (\log(n^2))$

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

(h)  $n^2 / \log n = \text{-----} (n \log^2 n)$

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

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

(k)  $\sqrt{n} = \text{-----} (\log^3 n)$

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

(m)  $n2^n = \text{-----} (3^n)$

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

(o)  $n! = \text{-----} (2^n)$

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

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

2. Work problem 0.3(c) on page 9 of the textbook.
3. Consider the following C++ program.

```
void process(int n)
{
    cout << n << endl;
    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?

4. The recursive algorithm implemented below as a C++ function is used as a subroutine during the calculation of the level payment of an amortized loan. What does it compute?

```
float squire(float x)
{
    return x*x;
}

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(squire(x),k/2);
}
```