University of Nevada, Las Vegas Computer Science 477/677 Fall 2022

Assignment 1: Due Tuesday September 5 2023 11:59 PM

Upload your homework to Canvas. Miss Wallace will explain how that is done.

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 September 2nd. Your file must not be unnecessarily long. If you have any questions, or you are having trouble uploading the assignment you may email the grader, Sebrina Wallace, at `wallace4@unlv.nevada.edu`. You may also send me email to ask questions.

1. True or False

   (a) _____ $n^2 = \Theta(n^3)$.

   (b) _____ $\log n^2 = \Theta(\log n^3)$.

   (c) _____ $\log^2 n = \Theta(\log^3 n)$.

   (d) _____ $n^{1.0001} = O(100 \log^{100} n)$

2. This problem is based on problem 0.3(c) on page 9 of the textbook. I have rewritten the problem, and the answer differs from the answer to the problem in the textbook.

   The Fibonacci numbers $F_1, F_2, \ldots$ are defined by the rules
   $F_1 = 1$
   $F_2 = 1$
   $F_n = F_{n-2} + F_{n-1}$ for $n \geq 3$

   (a) Compute F_n for $n = 3$, 4, 5, 6, 7, 8.

   (b) Find a constant C such that $F_n = \Theta(C^n)$?

3. The following code fragment takes $\Theta(n^2)$ time to execute. (We assume that n is given.)

```
for(int i = 0; i < n; i++)
  for(int j = 0; j < n; j++)
```

Here is a pseudocode version. It is supposed to be readable, but not in any particular programming language.

```
Read n
initialize count
increment count
i = 0
while i < n
   increment i
   increment count
   j = 0
   while j < n
      increment j
      increment count
Print count
```

(a) Write this program in C++.

(b) Run this program, entering various values of n, such as 1, 4, 10, 30, 100.

(c) Looking at the output, is it clear to you that the time complexity of the program is $\Theta(n^2)$?

Programming tip: never use an English word as an identifier in a program. Use a misspelled word, or a foreign word, instead. Why?

4. For each of these fragments:
   Expand the fragment to a C++ program with a counter.
   Run the code for $n = 1$, 10, 100, 1000, 1000000.
   For each, compute the final value of the counter for each n. Show the results in a table. (You do not have to turn in your code.)
   Guess the asymptotic time complexity, which will be $\Theta(1)$, $\Theta(\log n)$, $\Theta(n)$, $\Theta(n \log n)$, $\Theta(\sqrt{n})$, or $\Theta(n^2)$.

```
(a) for(int i = n; i > 0; i=i/2)
      i = i/2;

(b) for(int i = 1; i < n; i++)
      for(int j = n; j > i; j=j/2);

(c) for(int i = 0; i*i < n; i++)

(d) for(int i = 1; i < n; i++)
      for(int j = i; j > 0; j=j/2);
```

5. The recursive algorithm implemented below as the C++ function `mystery` computes a well-known algebraic operation. What is that operation?

```
float squre(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(squre(x),k/2);
 }
```