MATH 3336

HOMEWORK ASSIGNMENT 11

INSTRUCTIONS

- Record your answers to the following 10 questions. Show your work when a question requires you to do so.
- Scan your work and save the file as a .pdf (make sure your work and answers are legible)
- Upload your scanned work to CASA CourseWare using the "Assignments" tab. (Click this link for instructions on how to do this).
- Homework submitted after 11:59pm on the indicated due date will be assigned a grade of 0.
- Also, **DON'T FORGET THAT** $0 \in \mathbb{N}$.
- I understand that if any of the questions from this assignment (or future ones) are shared in ways that violate our Academic Honesty Policy, then the syllabus will change. Specificlaly, Homework and Quizzes will be worth zero points.

  Name:

  Signature:

1. (Part A) Consider the claim that "$1/x$ is $O(1)$." Is this true? If so, find "witnesses" k and C to explain why it is true. Is this claim false? If so, explain why.

(Part B) Consider the claim that "2^{n+1} is $O\left(2^n\right)$." Is this true? if so, find "witnesses" k and C to explain why it is true. Is this claim false? If so, explain why.

2. Explain why x^3 is $O\left(x^4\right)$ and that x^4 is *not* $O\left(x^3\right)$.

3. Explain why / prove that $5x^3 + 2x + 1$ is $\Theta\left(x^3\right)$.

4. Explain why / prove the following:

$$\text{If } f(n) \text{ is } O\left(\log_a(n)\right) \text{ then } f(n) \text{ is also } O\left(\log_b(n)\right)$$

where a and b are real numbers larger than 1.

(Hint: you may want to use this old algebraic fact: $\log_b x = (\log_a x)/(\log_a b)$)

5. Consider the following psuedocode for a linear search algorithm, one designed to determine where in a set (or list) of integers $\{a_1, a_2, \cdots, a_n\}$ a given number x may be located.

Algorithm 1 Linear Search Algorithm

```
   $i = 1$
while ($i \leq n$ and $x \neq a_i$)
$i \mapsto i + 1$
if  $i \leq n$  then  return $i$
else  return 0
```

The larger the value of n, the more steps / operations this algorithm requires. Explain why the worst-case run-time for this linear search algorithm is $\Theta(n)$.

6. Consider the following psuedocode for a binary search algorithm, one designed to determine where in a set (or list) of integers $\{a_1, a_2, \cdots, a_n\}$ a given number x may be located (here we will assume that the number of elements, n, is a power of two; i.e. $n = 2^k$).

Algorithm 2 Binary Search Algorithm

```
$i = 1$
$j = n$
while $i < j$
    $m \mapsto \lfloor (i + j)/2 \rfloor$
    if $x > a_m$ then $i \mapsto m + 1$
    else $j \mapsto m$
if $x = a_i$ then return $i$
 else return 0
```

The larger the value of n, the more steps / operations this algorithm requires. Explain why the worst-case run-time for this binary search algorithm is $\Theta(\log n)$ (and given the result of question 4, you can assume the logarithm here is base-2).

(You may first want to practice using this algorithm on a set like $\{4, 5, -10, 3, 7, -2, 8, 9\}$ and different values of x.)