CSC 611 – Algorithms and Advanced Data Strcutures
Exam #1, Fall 2024

First/Given Name: ____________________________

Last/Family Name: ____________________________

This exam contains 5 pages (including this cover page) and 3 questions.

- Clearly identify your answer for each problem, and try to organize your work in a reasonably coherent way, in the space provided. If you decided to use the back of a paper, note this clearly so the instructor can find your answer. You can simplify and shortify answers by combining terms, reducing fractions, etc, to an extent that it still shows you understand what are you doing.
- It might be a good idea to draw a box around your final answer.
- Unsupported final answers will not receive full credit, even if they are correct. Show your work to convince your instructor that you understand the material. Partial credit will be given for incorrect answers that show a partial understanding of the relevant concepts. Irrelevant and meaningless answers will not receive partial credit.
- No electronic devices, including calculators, are allowed.
- You have 30 minutes to complete this exam.
- Each student is allowed to use only a cheat sheet of size 4.5”×5.75”, which is equivalent to one-fourth of a standard letter size. The cheat sheet can be used on both sides. Only hand-written cheat sheets are allowed, and each student is required to write their name on their cheat sheet. The cheat sheet must be submitted along with the exam upon completion.

Question	Points	Score
1	2.00	
2	2.00	
3	2.00	
Total:	6.00	

I acknowledge that it is the responsibility of every student at Missouri State University to adhere to the university’s policies on Student Academic Integrity. I confirm that I have neither given nor received any unauthorized assistance during this exam.

Signature: ____________________________

(1) $\lim_{n\to\infty} a^n = 0 \quad (\forall a \in \mathbb{R} : 0 < a < 1)$

(2) $d+2d+3d+\cdots+nd = \frac{n(n+1)d}{2} \quad (\forall d \in \mathbb{R})$

(3) $\lim_{n\to\infty} a^n = \infty \quad (\forall a \in \mathbb{R} : a > 1)$

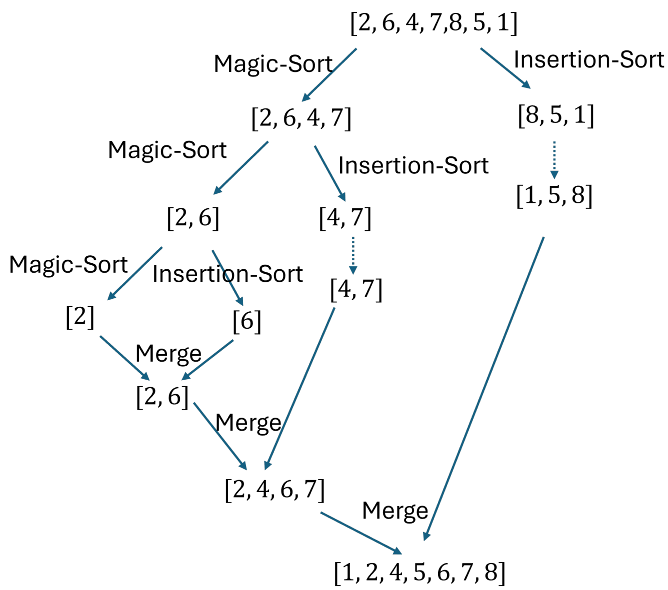
(4) $1 + r + r^2 + \cdots + r^n = \frac{1-r^{n+1}}{1-r} \quad (\forall r \in \mathbb{R} : r \neq 1)$

1. Consider a variant of Merge Sort, which we call Magic Sort, in which the left subarray is sorted recursively using Magic Sort, and the right subarraw is sorted using the Insertion Sort algorithm. The pseudocode for this algorithm is as follows.

```
MAGIC-SORT(A, l, r)
if l ≥ r                          // zero or one element
    return
m = ⌊(l + r)/2⌋                   // midpoint of A[l : r]
MAGIC-SORT(A, l, m)               // sort A[l : m] recursively
INSERTION-SORT(A, m + 1, r)       // sort A[m + 1 : r] using the insertion sort algorithm
MERGE(A, l, m, r)                 // merge A[l : m] and A[m + 1 : r] into A[l : r]
```

(a) (1.00 points) Illustrate the operations of MAGIC-SORT for the array $A = [2, 6, 4, 7, 8, 5, 1]$. You need to draw a tree showing what functions are called and what subarrays are passed to them. You do not need to show the operations performed within INSERTION-SORT and MERGE.

Solution: This question is very similar to the example illustrated in Figure 2.4 of the textbook. We also did another example in class. The only difference here is that the right subarray is assumed to be sorted using insertion sort.



Note that the question says not to illustrate the operations of INSERTION-SORT, so we represented the sorting of a subarray by INSERTION-SORT by a dashed edge.

 Points earned for this question: ______

(b) (0.50 points) Let $T(n)$ denote the running time of MAGIC-SORT for an array of size n and $S(n)$ be the running time of INSERTION-SORT for an array of size n. Provide a recurrence relation for $T(n)$.

Solution:

$$T(n) = \begin{cases} c_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + S(\lfloor n/2 \rfloor) + c_2 + c_3 n & \text{if } n > 1 \end{cases}$$

Note that if n is even, then both the left and the right subarrays have size $n/2$. If, n is odd, then the size of the left subarray is $(n+1)/2$, while the size of the right subarray is $(n-1)/2$. The ceiling and the floor in the above recurrence account for this fact.

(c) (0.50 points) If the array has already been sorted in non-decreasing order, how will the recurrence relation you provided in Part b look?

Solution: If the array has already been sorted, then, the insertion sort takes a linear time. Hence, $S(\lfloor n/2 \rfloor) = c_4(n/2) + c_5$. Accordingly,

$$T(n) = \begin{cases} c_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + c_4 \lfloor n/2 \rfloor + c_5 + c_2 + c_3 n & \text{if } n > 1. \end{cases}$$

This is simplified as

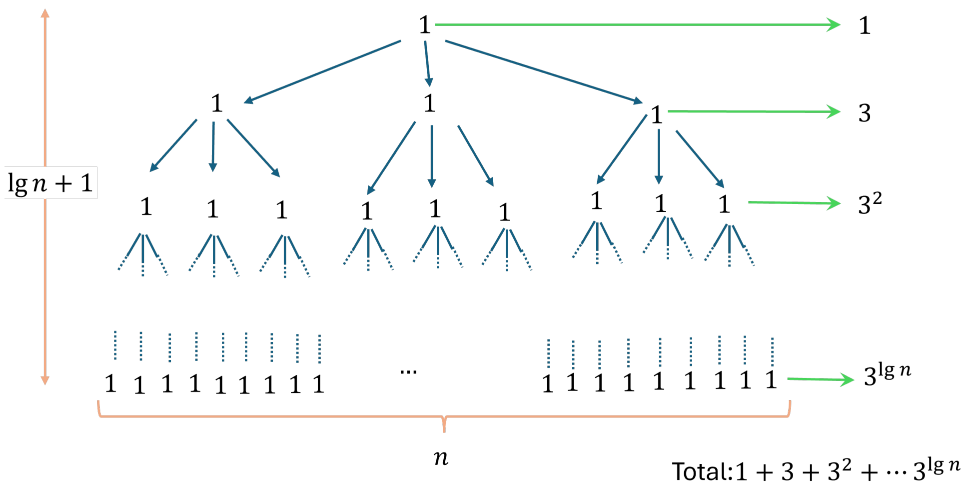
$$T(n) = \begin{cases} c_1 & \text{if } n = 1 \\ T(\lceil n/2 \rceil) + c_2' + c_3' n & \text{if } n > 1. \end{cases}$$

 Points earned for this question: ______

2. (2.00 points) Solve the following recurrence for the case where n is an exact power of 2.

$$T(n) = \begin{cases} 1 & \text{if } n = 1 \\ 3T(n/2) + 1 & \text{if } n > 1 \end{cases} \tag{1}$$

Solution: This question is very similar to Questions 1.5 and 1.6 from the first homework assignment. There are several methods to solve recurrences. Question 1.5 asks to expand the recurrence and use induction. In Question 1.6, you were asked to use the recursion tree method. With this method, the problem is solved as follows.



$$T(n) = 1 + 3 + 3^2 + \cdots 3^{\lg n} = \sum_{i=0}^{lgn} 3^i = \frac{1 - 3^{\lg n+1}}{1-3} = -\frac{1}{2} + \frac{3^{\lg n+1}}{2} = -\frac{1}{2} + \frac{3}{2} \cdot 3^{\lg n}$$
$$= -\frac{1}{2} + \frac{3}{2} \cdot n^{\lg 3}.$$

 Points earned for this question: ______

3. (2.00 points) Maddie claims that $3^{2n+1} \in O(2^{3n+1})$.

   Is she right? YES NO

   Provide **proof** for your answer.

   Solution This question is very similar to Q.1.10 (Exercise 3.2-3 of the textbook). We also proved in class that $3^n \notin O(2^n)$. This problem is very similar but with a slight twist.

   Assume Maddie is right. In that case, according to the definition of Big-O, there must exist constants $c > 0$ and n_0 such that for all $n \geq n_0$,

$$\begin{aligned}
3^{2n+1} &\leq c2^{3n+1} \\
3^{2n} \cdot 3^1 &\leq c2^{3n} \cdot 2^1 \\
(3^2)^n \cdot 3 &\leq c(2^3)^n \cdot 2 \\
9^n \cdot 3 &\leq c8^n \cdot 2 \\
\frac{3}{2} \cdot \frac{9^n}{8^n} &\leq c \\
\frac{3}{2} \cdot (\frac{9}{8})^n &\leq c \qquad (2)
\end{aligned}$$

   But, because $\frac{9}{8} \geq 1$, $\lim_{n\to\infty}(\frac{9}{8})^n = \infty$. Accordingly, the left side of (2) is unbounded. But, the right side of (2) is bounded because it is the constant c. This is a contradiction. Hence, the assumption that $3^{2n+1} \in O(2^{3n+1})$ is incorrect.

   The following approach to solve this problem might be even easier.

   Assume $f(n) = 3^{2n+1}$ and $g(n) = 2^{3n+1}$.

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{3^{2n+1}}{2^{3n+1}} = \lim_{n\to\infty} \frac{3}{2} \cdot \frac{9^n}{8^n} = \frac{3}{2} \lim_{n\to\infty} (\frac{9}{8})^n = \frac{3}{2} \cdot \infty = \infty \qquad (3)$$

   This proves that $f(n) \in \omega(g(n))$, meaning that f grows strcitly faster than g, and hence, $f(n) \notin O(g(n))$. Therefore, Maddie's claim is not correct.

Points earned for this question: ______