CS 692 Capstone Exam Algorithms Fall 2022. **Choose any 2 of the 3 problems.**

Full name: __________________ Net ID:__________________

Question 1) (10 points each case)

A. Consider the following recurrence relation and solve it to come up with a precise function of n in closed form (that means you should resolve all sigmas, recursive calls of the function T, etc.). **An asymptotic answer is NOT acceptable**. Justify your solution and show all your work.

$T(n) = T(n/2) + 2n$ where $T(1) = 1$ and $n = 2^k$ for a non-negative integer k.

B. Count the precise number of "fundamental operations" executed in the following code. Again, your answer should be a function of n ($n \geq 0$) in closed form. No asymptotic bound is accepted.

```
for(int i = 0; i <= n; i++)
{
        Perform 1 fundamental operation;
        for(int j = i+1; j <= n; j++)
                Perform 1 fundamental operation;
        //endfor j
}//endfor i
```

Question 2) (5 points each case)

Which of the following five statements correctly describes the relationship between the functions f and g defined in A)-D) below? Note that more than one of the five statements may be correct for each part. You do NOT need to justify your choices.

$f \in o(g)$ $f \in O(g)$ $f \in \theta(g)$ $g \in o(f)$ $g \in O(f)$

A) $f(n) = n!$, $g(n) = (n+1)!$

B) $f(n) = 2^n$, $g(n) = n^n$

C) $f(n) = 10n + \log(n^3) + 4$, $g(n) = \log n + 12$

D) $f(n) = \left\{ \begin{array}{ll} n^3 - 2n, & n \leq 10 \\ 3n^2 + 5, & n > 10 \;\; and\ n\ is\ odd \\ 12n^2, & n > 10 \;\; and\ \ n\ is\ even \end{array} \right\}$, $g(n) = 12n^2$

Question 3) (C/C++ coding question)
Write a **recursive** function which is provided an integer key value and a pointer to the root of a (possibly empty) binary tree, and searches for the key value in the tree. The tree should be implemented using **linked lists** and is to store n integer numbers.

A) **(2 points)** Declare your data structure.

B) **(8 points)** Code the search function as described above (no point for non-recursive function). Analyze the time complexity of your search function in the worst-case. Explain your answer and justify in detail.

C) **(10 points)** Now assume your given tree is a Binary Search Tree (BST). Write the search function again but this time consider the property of Binary Search Trees.

   Analyze your new algorithm for worst-case time complexity and discuss whether your algorithm in part (C) is more efficient than the one in part (B) or not in the following cases:
   (i) If the BST is balanced.
   (ii) If the BST is unbalanced.