CS 692 Algorithms Capstone Exam, Fall 2023.

Please read the following instructions before you start the exam. Please avoid asking a question that is addressed below:

1) You can write your answers **on both sides of the pages**. If you run out of space as you answer the questions on the front pages, just continue your answer on the back of the page. Please do not ask the instructor whether you can write on the back side or not again. The answer is YES YOU CAN.

2) You may use the **last paper** sheet (front and back) as the scratch paper. It is also marked as scratch paper on both sides. Please note that the instructor will NOT grade the scratch paper. The scratch sheet stays with the instructor. Please do not take that apart.

3) Do NOT disjoint the exam sheets. If at any point the papers got disjointed, raise your hand and request for stapling them immediately.

4) You are **NOT** allowed to use Calculator.

5) You are NOT allowed to have your cell phone, e-watch, or other electronic devices nearby like on the desk or inside your pocket. They should be turned off and stay inside your bag /backpack.

6) Exam duration: from 3:00 to 4:30 pm.

Please indicate the questions you have completed for grading. If not, we will assume that the first two questions you attempted are to be graded.

☐ Question 1

☐ Question 2

☐ Question 3

Choose any 2 of the 3 problems.
If you answer all three questions, only questions 1 and 2 will be graded.

Full name: _________________ Net ID:_________________

Question 1) (20 points)
Count the precise number of "fundamental/basic operations" executed in the codes below. Your answer should be a function of n ($n \geq 1$) in closed form. Note that "closed form" means that you must resolve all $\sum$'s, recursive relations, etc. An asymptotic answer (such as one that uses Big-O, Theta, and similar) is not acceptable.

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

B)
```
void fun(int n)
{
        Perform 1 fundamental/basic operation;
        if (n >0)
        {
                Perform 2n fundamental/basic operations;
                fun (n/2);
        }
}
```

Full name: __________________ Net ID:__________________

Full name: ________________ Net ID:________________

Question 2) (20 points)
Which of the following seven statements correctly describes the relationship between the functions f and g defined in A)-C) below? Note that more than one of the seven statements may be correct for each part. You do NOT need to justify your choices.

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

A) $f(n) = n!$, $g(n) = n^n$

B) $f(n) = 7 \log (n^4) + 4$, $g(n) = \log n + 12$

C) $f(n) = 2n+5$, $g(n) = 20$

D) $f(n) = \log_2 n$, $g(n) = \sqrt{n}$

Full name: ___________________ Net ID:___________________

Full name: _________________ Net ID:_________________

Question 3) (C/C++ coding question) (20 points)

Part a) (12 points) Write a complete C++ program with a Magic function that receives two binary **min heaps** as arrays T_1 and T_2, where T_1 has n nodes and T_2 has m nodes. Each node will be assigned different and distinct integer labels, meaning no two nodes in trees share the same label. The Magic function should generate a new array T_3 that is a binary **min heap** constructed out of values present in T_1 and T_2.

Note #1: Your code is permitted to utilize auxiliary space up to O(n+m).

Note #2: Ensure that your code operates with two minimum heaps and constructs a minimum heap from them. No credit will be awarded for code related to maximum heaps.

Note #3: Only C/C++ programming is accepted; any other language will receive no points.

Note #4: Implement binary trees in your code using basic (static) arrays as the underlying data structure.

Note #5: When executing the program, the user should input distinct and unique label values for the trees.

Note#6: You are not permitted to utilize pre-existing routines that perform the specific tasks we have asked you to do. In other words, using built-in libraries for tasks like array manipulation and heap construction is not allowed. Instead, you are expected to implement these operations from scratch in your code.

Part b) (6 points) Analyze the time complexity of your program in the worst case, step by step. Then, express its overall worst case time complexity using Big-O notation. Explain and show all your work.

Part c) **(2 points)** How did you ensure the auxiliary space complexity of your program is O(n+m). Briefly justify.

Full name: ___________________ Net ID:___________________

Full name: __________________ Net ID:__________________

Full name: ________________   Net ID:________________

Full name: ____________________ Net ID:____________________

Full name: __________________ Net ID:__________________

Full name: ___________________ Net ID:___________________

Scratch Page- This page (front and back) will not be graded.

Full name: __________________ Net ID:__________________