Your name: ________________________________

CS 3310 Midterm 1
Instructor: Gregory Mortensen

Instructions:

- *Do not discuss the exam with any other person, either within or outside the class.*
- *This exam is open book, open notes, open learning suite/slides, open internet. You must cite any sources you use (from the internet, your textbooks, etc.).* ***It is closed to discussion with friends, other students, online forums, etc.***
- Write your name on the top of every page you turn in since pages may be separated during grading (including scratch paper).
- Write your solutions in the space provided.
    - If you need more space, please write on the back of the sheet containing the problem.
    - Please do not write the solution on the back of another page of the exam, although you may attach additional scratch pages as necessary.
- There is no time limit; however, plan your time wisely. Do not spend too much time on one problem at the expense of others. Read through all the problems first and attack them in the order that allows you to make the most progress. There are 110 points possible the grade is out of 100 points. Therefore, you can earn up to 10% extra credit.
- ***Please show all of your work.*** You will be graded not only on the correctness of your answer, but on the clarity with which you express your rationale for your answer. Be neat! It is your job to make your understanding clear to us.
- ***You will not post or share this exam with anyone else -- by any means (physical, electronic, etc.)***
- If you find an error on or have a question about the exam, please teams me, thanks!

I agree to abide by the terms outlined above.

________________________________
(Your name)

________________________________
(Date)

Your name: ___________________________________

Question 1: **Big-O Proofs** (10 points)

Using the formal definition of Big-O and Omega **prove or disprove** the following. You need to do more than just give the necessary constants. You need to use the definition and go through all steps.

1. $7n^3 + 15n$, Omega($n^3$)

2. $20n^2 - 3n$ O(n)

Your name: ________________________________

Question #2 **Basic Operations** (20 pts)

A. What is the basic operation of the following code? Be precise. (5 pts)

```
accumulator: int = 0
n: int

for a:  1..n
   for b: 1..n^2
      for c: 1..n^3
         if (accumulator < a + b + c) then:
            add a, b, and c to the accumulator

return accumulator
```

B. What is the time complexity of the above algorithm? Why? (5 points)

C. Analyze complexity of Quick Sort (Best, Worst and Average cases). Remember to discuss both the actual best and worst cases closed form solutions ***in addition to it's big-Theta***). (10 pts)

Your name: ________________________________

Question #3 **Recurrence Relations**: (15 pts)
A. What is a recurrence relation? (5pts)

B. Solve the following recurrence (n>1, 10 points)

***Recurrence Relation:*** $t_n = t_{n-1} + \frac{1}{n*(n+1)}$

***Initial Constraint:*** $t_1 = \frac{1}{2}$

***Using a few values for n, determine the candidate solution:***

***Using induction, prove the candidate solution:***

Your name: ___________________________________

Question #4: **Dynamic Programming** (20 pts)

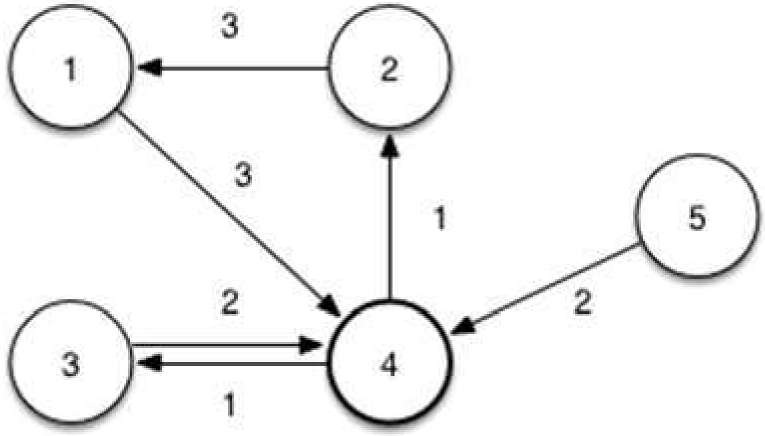
A: What is the Principle of Optimality (5 pts)

B: Does the Principle of Optimality apply to find the $n^{th}$ Fibonacci number? When and why? Or Why not? (10 pts)

C: Why is a dynamic programming solution of the Fibonacci solution better than a recursive divide and conquer solution? (5 pts)

Your name: ________________________________

Question #5: **Dynamic Programming** (15 pts)




**Note: You get the points if you trace the algorithm, just using the adjacency matrix, and not using the figures.**

Consider the above graph, use Floyd's algorithm for finding the shortest paths among all pairs of nodes. ***You don't need to write out each step.*** Just write the adjacency matrix from $k_1$ to $k_5$ ***showing only nodes that change during that step*** in the matrix as each iteration of k is applied. For example, if $k_1$ only changes the distance from node 2 to 3 from infinity to 1, but no other values in the matrix change, only fill in the cell down-2, across-3 (2,3) with 1. If the nodes are all blank for k2, then you wouldn't fill in any nodes for $k_2$.

Note: origination nodes are on the right margins; destination nodes are across the top.

The initial adjacency matrix (for $k_0$):

| $W_{i,j} = D^{(0)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | 0 | ∞ | ∞ | 3 | ∞ |
| 2 | 3 | 0 | ∞ | ∞ | ∞ |
| 3 | ∞ | ∞ | 0 | 2 | ∞ |
| 4 | ∞ | 1 | 1 | 0 | ∞ |
| 5 | ∞ | ∞ | ∞ | 2 | 0 |

| $W_{i,j} = D^{(1)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

Your name: ___________________________________

Q5 (Floyd), cont.

| $W_{i,j} = D^{(2)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

| $W_{i,j} = D^{(3)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

| $W_{i,j} = D^{(4)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

| $W_{i,j} = D^{(5)}_{i,j}$ | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| 1 | | | | | |
| 2 | | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |

Your name: ________________________________

Question #6: **Sequence Alignment** (20 pts)

A. Write the pseudocode for the gene sequence alignment algorithm(s). Remember to properly use [] for arrays and () for function invocations.

   1. Divide and Conquer version (5 pts)

   2. Dynamic Programming version (5 pts)

Your name: ________________________________

Q6 (Sequencing) cont.

B. In concise language (using only English, not Math), analyze the time and space complexities of both the DP and the D&C algorithms by comparing and contrasting the algorithms' time and space complexities, including why their respective time and space complexities are what they are. (10 pts)

Your name: ___________________________________

Q7 Homogeneous Linear Equations (10 pts, extra credit)

A. (2 pts) What is the characteristic equation of the recurrence (remember show all work):

$$t(k) = t(k-1) - 6t(k-2)$$

B. (4 points) Solve the recurrence given the initial constraints, $t(0) = 1$, $t(1) = 2$: ***(please show all work)***:

$$t(k) = t(k-1) - 6t(k-2)$$

Your name: ___________________________________

Q7 Cont.

C. (4 pts) Prove using induction that the candidate solution is correct