**CSCI 3302 Pair Programming Assignment 04 (100 Points)**

**Due: Oct 27, 8:00 AM**

**GitHub Link: NONE**

**Objectives:**

* Demonstrate understanding of algorithm analysis and Big-.
* Demonstrate how to compare algorithms experimentally.

**Assignment Assistance:**

* This homework assignment is due before the date and time specified above.
* You and your partner must work on this TOGETHER.
* This assignment is restricted to you and your partner. You may not receive help from any other person except the instructor or the AARC (help from the AARC must be well documented!).
* Any resource used (other than Dr. Becnel or the course text) must be documented in the code (as comments) detailing the source and describing exactly what was learned and how that information was used. Submissions will be severely penalized if copied in part or in whole from any source.
* If you need help, visit your instructor during his posted office hours. If your schedule cannot accommodate any of these times, then email your instructor to schedule a different time.

**Instructions:**

Answer the following questions. Show work and include explanations where appropriate. You can handwrite or type your answers. To type equations in MS Word, use Alt + =.

1. List the following from slowest growth rate to highest growth rate.
2. Using the results from class find the most appropriate for the following
3. Exactly how many times will “Hello” print in the following? You can check your answer by running the code block with an added count variable but should make a hypothesis first and have accompanying work showing how you arrived at the hypothesis.
   1. 
   2. 
   3. A picture containing text, font, screenshot, typography

      Description automatically generated
   4. A picture containing text, font, screenshot, number

      Description automatically generated
4. Find the most appropriate growth rate for the following code blocks. Include accompanying work to justify your solution. Your final answer should be in terms of , for example, :
   1. A picture containing font, text, screenshot, graphics

      Description automatically generated
   2. A picture containing text, font, screenshot, line

      Description automatically generated
   3. A picture containing text, font, screenshot, line

      Description automatically generated
   4. A picture containing text, font, screenshot

      Description automatically generated
   5. A computer code on a black background

      Description automatically generated with low confidence

1. Implement two versions for an algorithm to find the nth Fibonacci number. One should use recursion as we discussed previously in class. The other should use a loop. The loop version is below. Note: make sure to change the recursive version to use **long** and not **int** data types.

A picture containing text, screenshot, font

Description automatically generated

You will use both algorithms to compute the 10th, 20th, 30th, 40th, and 50th Fibonacci numbers. Create a table of your results, such as the one below.

|  |  |  |
| --- | --- | --- |
| N | Time to Compute Recursively | Time to Compute with Loop |
| 10 |  |  |
| 20 |  |  |
| 30 |  |  |
| 40 |  |  |
| 50 |  |  |

To measure the time each algorithm takes to execute use built-in Java timing mechanisms. Refer to this article (or similar) for assistance with Java timing mechanisms: [https://www.tutorialspoint.com/how-to-measure-execution-time-for-a-java-method](%20https://www.tutorialspoint.com/how-to-measure-execution-time-for-a-java-method)

Based on the table data, which algorithm is more efficient for computing Fibonacci numbers?

**Submission:**

* Only one person needs to turn in/commit the assignment. However, both members of the team are expected to understand the solutions and be able to answer questions about the solution.
* Review the Evaluation below to ensure you have met all the requirements.
* Put your and your partner's names on your answer sheet and upload it to D2L. Also, turn in your Java file from question 5.

**Evaluation**

* 1. Problem 1: 10 points; partial credit is given.
  2. Problem 2: 20 points (5 points each).
  3. Problem 3: 20 points (5 points each).
  4. Problem 4: 25 points (5 points each).
  5. Problem 5: 15 points (10 points for table; 5 points for correct answer).
  6. Miscellaneous: 10 points for neatness and following directions.