

Assignment 1

Deadline: Thursday, February 2, 2023, 11:59 pm

Objectives:

- Familiarize yourself with the basics of asymptotic analysis of algorithms
- Start designing algorithms to solve problems

Instructions:

You are not allowed to use advanced feature of Java such as **contains**. Use arrays of characters and loops. For problems marked as **(code)** – You should write a Java program that will be run by the instructor. You should provide a main method that calls your algorithm and prints to console both the input and output. For problems marked as **(text)** – You should write an answer. The answer can be either prose, an algorithm (pseudo-code or code), a proof, or mathematical equations. Whichever form of text answer is appropriate to solve the problem. In case the answer is an algorithm (code or pseudo code), it will be assessed mostly on logic rather than on whether it compiles.

Problems:

1. **(code) Common subsequence [30 points]**

A subsequence of a string is a new string generated from the original string with some characters (can be none) deleted without changing the relative order of the remaining characters.

For example, "ace" is a subsequence of "abcde".

Write an algorithm to solve the following problem:

Given two strings text1 and text2, return the length of their longest common subsequence. If there is no common subsequence, return 0.

Examples:

Input: text1 = "abc", text2 = "abc"

Output: 3

Explanation: The longest common subsequence is "abc" and its length is 3.

Input: text1 = "almanacs", text2 = "albatross"

Output: 3

Explanation: The longest common subsequence is "alas" and its length is 4.

Input: text1 = "almanac", text2 = "ferris"

Output: 0

Explanation: There is no such common subsequence, so the result is 0.

2. (Text) Common Substring [15 points]

A substring of a string is a new string generated from the original string without changing the order of the characters.

For example, "liar" is a subsequence of "peculiar".

Given two strings text1 and text2, write an algorithm that returns the longest common substring. If there is no common subsequence, return an empty string "". Case sensitive. Therefore, A and a are considered different characters. If there are multiple common substrings with the same length, your algorithm can return any of them.

Examples:

Input: text1 = "spy family", text2 = "jujutsu"

Output: ""

Explanation: There is no common substring between the two strings.

Input: text1 = "gears of war", text2 = "History of warriors"

Output: "of war"

Explanation: The longest common substring is "of war".

Input: text1 = "spy family", text2 = "jujutsu kaisen"

Output: "a" or "i" or "s"

Explanation: There are three possible common substrings between the two strings. In this case, your algorithm might return any of them

3. (code) Not Fibonacci [20 points]

Write a program that produces the NotFibonacci sequence, where the next number in the sequence is produced by averaging the previous two numbers. The rule for generating the NotFibonacci sequence is follows:

- Start with the initial values of 0 and 1.
- For each subsequent term, take the previous term, multiply it by 2, add it to the term before the previous, and append the result as the next term in the sequence.

$$n_i = (2 * n_{i-1}) + n_{i-2}$$

The program should be able to take an input of the number of terms in the sequence and produce the sequence up to that number. For example, if the user inputs 10, the program should output the first 10 numbers in the NotFibonacci sequence: 0, 1, 2, 5, 12, 29, 70, 169, 408, 985.

Hint: Use Long or BigInteger instead of int or Integer

4. (code) Where in Sequence [15 points]

Write a program to solve the following problem.

Given an integer output the position of that number in the NotFibonacci sequence from problem 3. If the given input is not a number in the sequence, return the position of the closest number lower than the input.

Examples:

Input: 8

Output: 6

Explanation: The number 8 is the 6th number in the NotFibonacci sequence.

Examples:

Input: 32

Output: 8

Explanation: The number 32 is not in the NotFibonacci sequence. Therefore, the closest lower number is 21 which is the 8th number in the NotFibonacci sequence.

5. (text) Algorithm Analysis [20 points]

For each of the algorithms you wrote for problems 1-4, explain their time complexity using Big-O, Big- Ω notation. Explain how you arrived at your answer.

Extra credit:

(text) Generate a plot of the first 1000 number in the NotFibonacci sequence [5 points]

Grading Rubric:

Item	Points
Common substring	30
Common subsequence	15
NotFibonacci	20
Where in Sequence	15
Algorithm analysis	20
Extra credit	5
Total Points	100

Deliverables:

A zip file containing (1) the code in an IntelliJIDEA project with the implementation of the (code) problems and (2) a pdf with the answers to the (text) problems submitted on Blackboard. **Make sure you double-check that the file was uploaded correctly on Blackboard!** I will not be accepting excuses after the deadline that the file was not uploaded correctly.

You may submit a physical (by hand) solutions to (text) problems. Please be sure your work is written clearly and readable.