

OOP Assignment 1

Release date - 07/02/2024

Deadline - 22/02/2024

OVERVIEW

The first programming assignment will involve working with large binary numbers and understanding running time behavior of various algorithms operating on these numbers.

As you know by now, Java is an Object Oriented Programming Language and one important feature of object oriented design is encapsulation. Here, we can create program components where data can be bundled together with methods that act on the data and the internal working of the component is private to the component. Since our goal in the first assignment will be to do manipulations on large binary numbers, we will use encapsulation and define a java class related to large binary numbers. Let us call this java class **myBinaryNumber**.

Task - 1 (10 marks)

Firstly you need to implement the class **myBinaryNumber**. Description of the class is as following -

- Constructor *myBinaryNumber(int n)*: This initializes an all 0's n-bit binary number.
- Constructor *myBinaryNumber(String S)*: This initializes a binary number represented by the string S. So, the length of the number is the size of the string.
- Public method *getSize()*: This returns the size of the binary number. That is the number of bits.
- Public method *getBit(int p)*: This returns the bit at position p from the least significant bit (LSB) side. That is if p = 0, it returns the LSB.
- Public method *setBit(int p, int b)*: This method sets the bit at position p (from LSB side) as b. You should throw an exception if the inputs are not valid. The programmer using this method should make sure to handle the exception.
- Public method *printNumber()*: This prints the binary number.

- Public method *printDecimalNumber()*: This prints the decimal equivalent of the binary number.

Task - 2 (30 Marks)

Now that we have represented a binary number using the class **myBinaryNumber**. We would like to do operations on binary numbers. So basically do operations on objects of the type **myBinaryNumber**. For this, you have to define an *abstract* class named **binaryOperations**.

In abstract class **binaryOperations**, you need to define two methods -

- Abstract method **binaryMultiplication(myBinaryNumber a, myBinaryNumber b)**: This method takes two binary numbers and returns their multiplication. This method should be implemented by any class which extends the class **binaryOperations**.
- Non abstract method **binaryAddition(myBinaryNumber a, myBinaryNumber b)**: This method takes two binary numbers and returns their addition. You should implement this method to compute the sum of two binary numbers and return the result. You should already know how to do binary addition from your previous courses but if you don't know then there are lots of resources available on the internet for example this youtube video ([Gate Smasher Binary Addition](#)) from Gate Smashers.

Task - 3 (30)

Now that we have implemented the binary addition, we would like to extend it to binary multiplication also. Your task is to create a class called **binaryMultiplicationNaive** which *extends* the class *binaryOperations*. In this class, you need to provide the basic/naive implementation for the method *binaryMultiplication*.

Task - 4 (30)

In this task, you need to create a driver code for your program to test whether your code is working correctly. You have to implement a class called **Driver** and inside that you need to create a main function. Your main function should read two binary numbers

from a text file and initialize the objects of *myBinaryNumber* class. Then it should compute the addition and multiplication of both numbers and write those inside a text file, where the first line of the output text file should be addition and second line should be the multiplication of the binary numbers.

Task - 5 (Bonus, 20 marks)

This is a bonus task so it's not compulsory to do it but if someone is able to do it then they can receive a **20 marks bonus** in the assignment therefore one can get 120 out of 100 marks in the assignment. **But this bonus option is only available if you have correctly completed the tasks 1 to 4, if you didn't complete tasks 1 to 4 then task-5 will not be evaluated even if you have implemented it.**

In this task, first you have to run your program on large inputs and compute the running time of the *binaryMultiplication* function for each input (Running time of only multiplication function, not entire program, You can do this using inbuilt functions of java which can compute time a system takes to run a piece of code.) and make a plot of running time (y-axis) and size of input n (x-axis). Your plot should look like a quadratic function as the time complexity of binary multiplication of n -bit numbers is $O(n^2)$. You can take $n = 10, 20, 30, 40, 50, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000$ for your plot.

Now you know that naive implementation is quite slow ($O(n^2)$). You might be thinking "can we do something better in terms of complexity", and the answer is "yes, we can". There is a quite popular algorithm called [the Karatsuba algorithm](#) which provides an approach to compute binary multiplication in just $O(n^{1.58})$ time.

Your second task is to create another class called **binaryMultiplicationFast** which *extends* the class *binaryOperations*. In this class, you need to provide the karatsuba algorithm implementation for the method *binaryMultiplication*.

And then perform the same analysis, you have done for naive implementation for the various size inputs.

Submission Instructions

- You can do this assignment in a group of at max 3 people.
- Your final submission should be a zip file named **<rollNo1>_<rollNo2>_<rollNo3>.zip** All your source code should be inside this zip file.
- Your implementation should be your own. It should not be copied from online resources and other teams.
- Since there is no way for me to determine which group copied from which one, I will give zero to all groups whose code is found similar.
- All groups are responsible for handling their source code. Excuses like *“my girlfriend/boyfriend/friend knows my laptop’s password and he/she stole the code without my consent”*, or *“I just shared with one friend, I don’t know how other groups got access to my code”* etc. won’t be tolerated.