

THE UNIVERSITY OF WESTERN ONTARIO

DEPARTMENT OF COMPUTER SCIENCE
LONDON CANADA

Analysis of Algorithms (Computer Science 3340b)

ASSIGNMENT 1

Due date: Thursday, February 2, 2023, 11:55 PM

1. **A complete binary tree** is defined inductively as follows. A complete binary tree of height 0 consists of 1 node which is the root. A complete binary tree of height $h + 1$ consists of two complete binary trees of height h whose roots are connected to a new root. Let T be a complete binary tree of height h . Prove that the number of leaves of the tree is 2^h and the size of the tree (number of nodes in T) is $2^{h+1} - 1$.
2. A Fibonacci-like number sequence is defined as follows: $N_0 = 2$; $N_1 = 2$; $N_n = N_{n-1} + N_{n-2}$, $n > 1$.
Prove that $N_n = 2F_{n+1}$, $n \geq 0$ where F_i , $i \geq 1$ is the Fibonacci number.
3. Exercise 2.3-5 (pp. 44) in the textbook.
4. Read the textbook for the definition of o and ω and answer Problem 3-2 (pp. 71) in the textbook.
5. Problem 4-6 b, c, and d (pp. 122-123) in the textbook.
6. Suppose that the running time of a recursive program is represented by the following recurrence relation:

$$\begin{aligned} T(2) &\leq 2c \\ T(n) &\leq 2T(n/2) + cn^2 \log_2(n) \quad n > 2 \end{aligned}$$

Determine the time complexity of the program using recurrence tree method (not using master theorem) and then prove your answer.

7. Consider the Fibonacci-like number sequence:

$$N_n = N_{n-1} + N_{n-2}, \quad n > 1; \quad N_1 = 2; \quad N_0 = 2.$$

- a) Write a recursive function to compute N_n using the above definition directly. Implement your solution and print N_{i*5} , where $0 \leq i \leq 10$, as output.
- b) Write a recursive function/procedure to compute N_n with time complexity $O(n)$ (more precisely, the time complexity should be $O(nA(n))$ when n is large, where $A(n)$ is the complexity of adding N_{n-1} and N_{n-2}). Implement your solution and print N_{i*20} , where $0 \leq i \leq 25$, as output. This program must be able to compute N_n precisely for $n \leq 500$.

Hint 1:

$$\text{Let } G_n = \begin{pmatrix} N_n \\ N_{n-1} \end{pmatrix} : G_n = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \times G_{n-1}, \quad n > 1; \quad G_1 = \begin{pmatrix} 2 \\ 2 \end{pmatrix}; \quad G_0 = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

With this formulation, design a recursive algorithm for G_n such that the algorithm will return both N_n and N_{n-1} with input parameter n .

Hint 2: Can you use a primitive type to store N_{500} ?

For programs in 7 a) and 7 b) of this question, you are **NOT** allowed to use Python. For C++ and Java, you can only use primitive types such as char, int, long and long long. You are not allowed to use large integer, such as BigInteger in Java, from the language library. You have to write your own large integer class or data type, if needed.

- c) Use the Unix time facility (bash time command) to track the time needed for each algorithm. Compare the results and state your conclusion in two or three sentences.
- d) Can you use your program in 7 a) to compute N_{50} if int type of 4 bytes is used? Briefly explain your answer. Explain why your program in 7 b) can compute N_{500} precisely?

Algorithms and answers for 7 b) to 7 d) should be in **asn1_solutions.pdf**

For this question, for C++, when compiling option O2 could be considered and a makefile should be written such that the command "make clean" will remove all the "*.o" files, the command "make asn1_a" will generate an executable file "asn1_a" for 7 a) that can be run by typing "asn1_a"; and the command "make asn1_b" will generate an executable file "asn1_b" for 7 b) that can be run by typing "asn1_b". If you are using Java, you may not need the makefile. In that case, you should have shell script files, "asn1_a" and "asn1_b" such that by typing, "asn1_a" and "asn1_b", your java programs will run.

You should use unix command "script" to capture the screen of the execution of your program.

Your programs have to be able to run on **compute.gaul.csd.uwo.ca** as our TAs will be marking your programs there.