Northeastern University
College of Engineering
Department of Electrical & Computer Engineering

EECE7205: Fundamentals of Computer Engineering

Spring 2020 - Homework 6

Instructions

- For programming problems:
  - Your code must be well commented by explaining what the lines of your program do. Have at least one comment for every 4 lines of code.
  - You are not allowed to use any advanced C++ library unless it is clearly allowed by the problem. For example, you cannot use a library function to sort a list of data if the problem is asking you to implement an algorithm to sort the list.
  - At the beginning of your source code files write your full name, students ID, and any special compiling/running instruction (if any).
  - Test your code on the COE Linux server before submitting it:
    a. If your program does not compile in the COE server due to incompatible text encoding format, then before uploading your source code file to the server make sure it is saved with Encoding Unicode (UTF-8). In visual studio, Save As -> Click on the arrow next to Save -> Save with Encoding -> Yes -> Unicode (UTF-8) -> Ok
    b. Compile using g++ -std=c++11 <filename>
- Submit the following to the homework assignment page on Blackboard:
  - Your homework report developed by a word processor (no handwritten or drawn contents are acceptable) and submitted as one PDF file. The report includes the following (depending on the assignment contents):
    a. Answers to the non-programming problems.
    b. A summary of your approach to solve the programming problems.
    c. A summary of the skills you acquired and challenges you faced by implementing the programs.
    d. Your recommendations of extension to the programming problems.
    e. The screen shots of the sample run(s) of your program(s)
  - Your well-commented programs source code files (i.e., the .cc or .cpp files).

  Do NOT submit any files (e.g., the PDF report file and the source code files) as a compressed (zipped) package. Rather, upload each file individually.

Note: Yon can submit multiple attempts for this homework, however, only your last submitted attempt will be graded.

Problem 1 (30 Points)

In a Fibonacci sequence, each number is the sum of the two preceding ones, starting from 0 and 1. The first 12 numbers of the sequence are: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, …

Write a C++ program to find the *n*th number in a Fibonacci sequence using two different approaches. Each approach is implemented by a function that takes *n* as its only parameter. One function solves the problem only recursively. The other function solves the problem recursively but with the support of dynamic programming. For the dynamic programming implementation, you are allowed to define the needed array as a global variable. You are not allowed to use any loop in your functions. The only loop you will need in your program is the one in the `main()` function, right before calling the dynamic version function. You need this loop to initialize all elements in the array to an invalid Fibonacci number (e.g., -1) to keep track of the array elements that do not have a Fibonacci number calculated yet.

Test your code with *n* values 5, 10, 20, 30, 40

Submit your test program and in your report answer the following questions:

a) Comment on the running time of each function.
b) For each function, find the big *O* asymptotic notation of its running time growth rate.

Problem 2 (40 Points)

Write a C++ program to compare the *Recursive* implementation of the *Rod Cutting* problem with the *Dynamic Programming* implementation of the same problem. The comparison should be in terms of each implementation running time for rod sizes: 5, 10, 15, …… , 40, 45, and 50 inches.

- Use the C++ clock function to measure the time consumed by each implementation in **micro-seconds**. (Reference: http://www.cplusplus.com/reference/ctime/clock/)
- Use the following formula to calculate the price of each cut of length *L* inches:

$$\text{Price} = \begin{cases} 2 & \text{if } L = 1 \\ \mathbf{floor}(L*2.5) & \text{if } 1 < L < \text{rod size} \\ (L*2.5)-1 & \text{if } L = \text{rod size} \end{cases}$$

where **floor** is a C++ function that returns the largest integral value that is not greater than the value of its parameter.
(Reference: http://www.cplusplus.com/reference/cmath/floor/)

Based on the results from your program, fill the following table:
(*Note*: if a run takes more than 2 minutes to finish, just terminate it and report in the table "no solution").

Rod Size	Recursive Time	Recursive Max Revenue	Dynamic Time	Dynamic Max Revenue
5				
10				
15				
20				
25				
30				
35				
40				
45				
50				

Problem 3 (30 Points)

Find the Huffman encoding tree and hence the Huffman code for each of the letter shown in the following letter-frequency table:

Letter	Z	K	M	C	U	D	L	E
Frequency	2	7	24	32	37	42	42	120