CSC 330 Assignment 2

Chapter 5:

Q1 (Written):
Textbook Page 217, In Short, 5.3
Estimated time: 2 minutes

Q2 (Written):
Textbook Page 217, In Short, 5.4
Estimated time: 2 minutes

Q3 (Written):
Textbook Page 217, In Short, 5.5
Estimated time: 5 minutes

Q4 (Written):
Textbook Page 217, In Short, 5.6
Estimated time: 2 minutes

Q5 (Written):
Textbook Page 217, In Short, 5.7
Estimated time: 2 minutes

Q6 (Written):
Textbook Page 217, In Short, 5.8
Estimated time: 10 minutes

Q7 (Written):
Textbook Page 217, In Theory, 5.10
Estimated time: 2 minutes

Q8 (Written):
Textbook Page 218, In Theory, 5.11
Estimated time: 5 minutes

Q9 (Written):
Textbook Page 218, In Theory, 5.14
Estimated time: 5 minutes

Q10 (Written):
Textbook Page 218, In Theory, 5.15
Estimated time: 5 minutes

Q11 (Written):
Textbook Page 219, In Theory, 5.19
Estimated time: 5 minutes

Q12 (Written):
Textbook Page 220, In Theory, 5.20 **a.** (note: **part a only**)
Estimated time: 10 minutes

Q13 (Written):
Textbook Page 221, In Theory, 5.21 **a.** (note: **part a only**)
Estimated time: 2 minutes

Q14 (Programming):
Textbook Page 224, In Practice, 5.33 (note: print the running time and related explanation as part of the output**)**
If you cannot achieve *O(N),* give an program/algorithm with your best running time.
Hint: Design the algorithm on paper first, before you start programming.
Estimated time: 1 hour to 2 hours

Chapter 7:

Q15 (Written):
Textbook Page 342, In Theory, 7.17 **b, e, h**

Q16 (Programming):
Textbook Page 343, In Practice, 7.23
Please provide some test inputs.
Hint: Input can be "long n"
Estimated time: 1 hour

Q17 (Written):
Maximum contiguous subsequence sum problem (textbook page 320).
Use the divide-and-conquer algorithm (textbook page 320 to 327), suppose the input is
{ -4, -4, 2, -1, 1, 2, 5, -2 }. Trace the recursive calls for the problem. Record how the maximum sum is chosen for each step (the three cases).
You can submit a hand-written copy for this question. If you submit a hand-written copy, you do not need to submit the answer of this question to blackboard.
Hint: Same as the binary tree we did in class.
Estimated time: 1 hour

Submission instructions:
Please submit a paper copy and an electronic copy.

Paper copy:

- Please submit the paper copy at the beginning of the class.
- Provide (create) **test input data** to all programming questions, and capture the related outputs as screen captures (or output files).
- Print the program source code files, test input data and **the output screen captures (or output files)**. If no output screen capture (or output file) is submitted, it would be assumed that the related program does not compile. If the print-out is not readable, **no mark will be awarded**.
- Identify each assignment question by writing the question number at the top of each page.
- Add the following statement to the first page of your submission: "I have abided by the UNCG Academic Integrity Policy on this assignment". Please write your full name and sign next to the statement. If the statement or the signature is not found, **75% of the possible points will be deducted**.

Electronic copy:

- Please submit a **lastname_firstname_assignment02.zip** (or lastname_firstname_assignment01.rar) file through the Blackboard Digital Dropbox. This zip (or rar) file should contain all submission files.
- Put the answers of all written questions in a **lastname_firstname_assignment02.doc** file.
- For programming questions, you only need to submit the *.java files.
- Name each *.java file (and related java class) according to the question number (e.g. Q1_*.java).

Grading guidelines (programming questions):
Your programs will be judged on several criteria, which are shown below.

- Correctness (50%): Does the program compile correctly? Does the program do what it's supposed to do?
- Design (20%): Are operations broken down into classes and methods in a reasonable way?
- Style (10%): Is the program indented properly? Do variables have meaningful names?
- Robustness (10%): Does the program handle erroneous or unexpected input gracefully?
- Documentation (10%): Do all program files begin with a comment that identifies the author, the contents, and the compiler used for that particular file? Are all the classes, methods and data fields clearly documented? Are unclear parts of code documented?

A program that does not compile will get at most **50% of the possible points**.