INSTRUCTIONS

Homework should be done in groups of one to three people. You are free to change group members at any time throughout the quarter. Problems should be solved together, not divided up between partners. A single representative of your group should submit your work through Gradescope. Submissions must be received by 11:59pm on the due date, and there are no exceptions to this rule.

Homework solutions should be neatly written or typed and turned in through Gradescope by 11:59pm on the due date. No late homeworks will be accepted for any reason. You will be able to look at your scanned work before submitting it. Please ensure that your submission is legible (neatly written and not too faint) or your homework may not be graded.

You may consult their textbook, class notes, lecture slides, instructors, TAs, and tutors for help with homework. You should not look for answers to homework problems in other texts or sources, including the internet. Only post about graded homework questions on Piazza if you suspect a typo in the assignment, or if you don’t understand what the question is asking you to do. Other questions are best addressed in office hours.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

For questions that require pseudocode, you can follow the same format as the textbook, or you can write pseudocode in your own style, as long as you specify what your notation means. For example, are you using “=” to mean assignment or to check equality? You are welcome to use any algorithm from class as a subroutine in your pseudocode.

READING Rosen Sections 3.1: pages 191-194 (up to but not including Searching Algorithms) and pages 198-201 (except proofs; we’ll come back to the proofs in future weeks), Appendix 3 pages A11-A16. Jenkyns and Stephenson Section 1.1 pages 1-2, Section 3.1 pages 77-79, Section 3.61 page 115 (just the pseudocode, the proof will come later).

KEY CONCEPTS Algorithm, input of algorithm, output of algorithm, tracing pseudocode given input, higher-level function of an algorithm, definiteness of algorithms, finiteness (termination) of algorithms, correctness of algorithms, optimization problem, greedy approach.
1. (2 points) Set up all the tools required for this class: Enroll in the Piazza site for this class: http://piazza.com/ucsd/spring2016/cse20. Sign up for a discussion section at sections.ucsd.edu. Register your iClicker at https://www1.iclicker.com/register-clicker/. Confirm you have access to this class via Gradescope https://gradescope.com/ using your ucsd.edu email address. Access the textbook website http://highered.mheducation.com/sites/0073383090/student_view0/index.html to find additional examples and self-test quizzes.

To receive credit for this question, submit screen-captures of each group member’s account logged into Piazza.

2. (12 points) Consider the following algorithm:

\[
\begin{align*}
\text{procedure } \text{alg1}(a, b : \text{ positive integers}) \\
x &:= a \\
y &:= b \\
\text{while } y \neq 0 \\
r &:= x \mod y \\
x &:= y \\
y &:= r \\
\text{return } x
\end{align*}
\]

(a) What does \text{alg1} return with input \(a = 50, b = 75\)? Include enough work to trace the execution of the algorithm on these inputs. Clearly label the output of the algorithm in your answer.

(b) Briefly explain why \text{alg1} is a finite algorithm.

Note: for full credit, cite explicitly the definition of a finite algorithm and then describe why the particular algorithm you are analyzing matches this definition. Your answer should make specific reference to line items in the pseudocode and their impact on the behavior of the algorithm.

3. (9 points) What is the value of \(x\) after each of these statements is encountered in a computer program, if \(x = 1\) before the statement is reached? Make sure to justify your answers.

(a) if \(x + 2 = 3\) then \(x := x + 3\)

(b) if \(x < 2\) then \(x := x \div 4\)

(c) if \(x \neq 1\) then \(x := 2\)

(Similar to Rosen 1.1 #42.)

4. (12 points) Show that if there were a coin worth 9 cents, the greedy algorithm using quarters, dimes, 9-cent coins, nickels, and pennies would not always produce change using the fewest coins possible.

Note: for full credit, you need to find an example where the greedy algorithm does not produce change using the fewest coin possible, and then describe why this example works (how many coins would the greedy algorithm produce in change? what's a better way to make change using fewer coins?)

Second note: for worked examples related to this question, refer to Week 1 discussion section worksheet.

(Rosen 3.1 #56.)
5. (15 points) Recall the Russian Peasant Multiplication algorithm

procedure RPM(m : real number; n : positive integer)

1. total := 0
2. a := m
3. b := n
4. while b > 1
5. if (b mod 2 = 1) then total := total + a
6. a := 2 · a
7. b := b div 2
8. return total + a

(a) Trace (i.e. walk through) an example of the operation of this algorithm when m = 208 and n = 41. Record the values of all variables during this trace.

(b) Briefly explain why RPM is a finite algorithm.

Note: for full credit, cite explicitly the definition of a finite algorithm and then describe why the particular algorithm you are analyzing matches this definition. Your answer should make specific reference to line items in the pseudocode and their impact on the behavior of the algorithm.