Timothy\_Zwart\_CMPT\_435

Assignment 10

**Date Assigned: 11/16/2018**

**Due: Midnight 11/30/2018 on iLearn**

**Please read turn-in checklist at the end of this document before you start doing exercises.**

**Section 1: Pen-and-paper Exercises**

1. Suppose we want to make change for N cents, using the least number of coins of denominations {1, 10, 25} cents (Different from the US currency system!). Consider the following greedy strategy: suppose the amount left to change is M, take the largest coin that is no more than M; subtract this coin's value from M, and repeat.

Does this algorithm output an optimal solution? If not, give a counterexample. If yes, prove that this algorithm always outputs an optimal solution (a formal proof as what we have done in the video).

**5 points – Your answer**

**The algorithm does not provide an optimal solution.**

**10 points – Counterexample or proof**

**Here is a counterexample to show that greedy algorithm is not the most optimized algorithm for this problem.**

**{1, 10, 25}**

**M=57**

**Greedy: 25+25+1+1+1+1+1+1+1 = 9 coins total**

**Optimal: 25+10+10+10+1+1 = 6 coins total**

**As you can see, there exists a more efficient algorithm that will allow us to get the same number of change with less coins.**

**Section 2: Java Implementation**

1. Implement the Greedy Coin Changing Algorithm discussed in the video in Java.

Note:

Find a file called Coinchange.java in assignment 10 folder.

Complete the method of greedycoinchange().

Test your method in the main method provided following the comments.

**Important: In all of the assignments of this course, when you are asked to implement an**

**algorithm for a problem, your code will be evaluated based on:**

**5 points - Execution**

**Each file must run without error or warning on valid input described in the main method provided.**

**5 points - Within Code Documentation**

**Is the code documented for obvious understanding of the use, preconditions, and postconditions of each function?**

**20 points - Correctness**

**Is the algorithm implemented correctly? Does your method pass the test?**

**TURN-IN CHECKLIST:**

1. **Answers to Section 1 (.doc/.txt/.pdf), and to Section 2 (all your source Code (.java files)). Remember to include your name, the date, and the course number in comments near the beginning of your code/report.**
2. **Create a folder and name it 'FirstName\_LastName\_assignment\_10'. In the newly created folder copy and paste your files (.doc/.txt/.java files). Then compress the folder, and push it to iLearn.**