Homework #1, CpSc 3120 Spring 2018 (Due: March 25)

A palindrome is a string that reads the same from front and back. Any string can be viewed as a sequence of palindromes if we allow a palindrome to consist of one letter.

Example: "bobseesanna" can be viewed as being made up of palindromes in the following ways:
"bobseesanna" = "bob" + "sees" + "anna"
"bobseesanna" = "bob" + "s" + "ee" + "s" + "anna"
"bobseesanna" = "b" + "o" + "b" + "sees" + "a" + "n" + "n" + "a"

We want to compute a function MinPal(s) defined as the minimum number of palindromes from which one can construct s (that is, the minimum k such that s can be written as $w_1w_2 \ldots w_k$ where $w_1, w_2, \ldots, w_k$ are all palindromes). Example: MinPal("bobseesanna")=3 since "bobseesanna" = "bob" + "sees" + "anna" and we cannot write "bobseesanna" with less than 3 palindromes.

(a) Consider an arbitrary string s made up from the alphabet "a" through "z" (no other symbols and no blanks). Design a Divide and Conquer algorithm to compute MinPal(s). You will write a complete paragraph explaining the principle of your algorithm (provide the recursive formulation you use). Then write a complete pseudocode of your algorithm with enough comments and declarations of the data structures you use.
(b) Show that the running time of MinPal(s) is exponential in the length n of s.
(c) Design a Dynamic Programming $O(n^3)$ algorithm to solve the problem (show that your algorithm is $O(n^3)$); write a program to implement and show experimental results.

Hint: The problem is similar to the matrix chain product problem we detailed in class.

How to Submit: Generate a single pdf file including everything in an organized way with headings and email to psriman@clemson.edu with Subject Line … CpSc 3120 Homework #1, <Your Namme>.