1. Suppose we are given an array $A[1..n]$ of integers, some positive and negative, which we are asked to partition into contiguous subarrays, which we call ***chunks***. The *value* of any chunk is the *square* of the sum of elements in that chunk; the value of a partition of A is the sum of the values of its chunks.

   For example, suppose $A = [3, -1, 4, -1, 5, -9]$. The partition $[3, -1, 4], [-1, 5], [-9]$ has three chunks with total value $(3-1+4)^2 + (-1+5)^2 + (-9)^2 = 6^2 + 4^2 + 9^2 = 133$, while the partition $[3, -1], [4, -1, 5, -9]$ has two chunks with total value $(3-1)^2 + (4-1+5-9)^2 = 5$.

   (a) Describe and analyze an algorithm that computes the minimum-value partition of a given array of n numbers.

   (b) Now suppose we also given an integer $k > 0$. Describe and analyze an algorithm that computes the minimum-value partition of a given array of n numbers ***into at most k chunks***.

2. Consider the following solitaire form of Scrabble. We begin with a fixed, finite sequence of tiles; each tile contains a letter and a numerical value. At the start of the game, we draw the seven tiles from the sequence and put them into our hand. In each turn, we form an English word from some or all of the tiles in our hand, place those tiles on the table, and receive the total value of those tiles as points. If no English word can be formed from the tiles in our hand, the game immediately ends. Then we repeatedly draw the next tile from the start of the sequence until either (a) we have seven tiles in our hand, or (b) the sequence is empty. (Sorry, no double/triple word/letter scores, bingos, blanks, or passing.) Our goal is to obtain as many points as possible.

   For example, suppose we are given the tile sequence

   $$\boxed{I_2}\,\boxed{N_2}\,\boxed{X_8}\,\boxed{A_1}\,\boxed{N_2}\,\boxed{A_1}\,\boxed{D_3}\,\boxed{U_5}\,\boxed{D_3}\,\boxed{I_2}\,\boxed{D_3}\,\boxed{K_8}\,\boxed{U_5}\,\boxed{B_4}\,\boxed{L_2}\,\boxed{A_1}\,\boxed{K_8}\,\boxed{H_5}\,\boxed{A_1}\,\boxed{N_2}\,.$$

   Then we can earn 68 points as follows:

   - We initially draw $\boxed{I_2}\,\boxed{N_2}\,\boxed{X_8}\,\boxed{A_1}\,\boxed{N_2}\,\boxed{A_1}\,\boxed{D_3}$.
   - Play the word $\boxed{N_2}\,\boxed{A_1}\,\boxed{I_2}\,\boxed{A_1}\,\boxed{D_3}$ for 9 points, leaving $\boxed{N_2}\,\boxed{X_8}$ in our hand.
   - Draw the next five tiles $\boxed{U_5}\,\boxed{D_3}\,\boxed{I_2}\,\boxed{D_3}\,\boxed{K_8}$.
   - Play the word $\boxed{U_5}\,\boxed{N_2}\,\boxed{D_3}\,\boxed{I_2}\,\boxed{D_3}$ for 15 points, leaving $\boxed{K_8}\,\boxed{X_8}$ in our hand.
   - Draw the next five tiles $\boxed{U_5}\,\boxed{B_4}\,\boxed{L_2}\,\boxed{A_1}\,\boxed{K_8}$.
   - Play the word $\boxed{B_4}\,\boxed{U_5}\,\boxed{L_2}\,\boxed{K_8}$ for 19 points, leaving $\boxed{K_8}\,\boxed{X_8}\,\boxed{A_1}$ in our hand.
   - Draw the next three tiles $\boxed{H_5}\,\boxed{A_1}\,\boxed{N_2}$, emptying the list.
   - Play the word $\boxed{A_1}\,\boxed{N_2}\,\boxed{K_8}\,\boxed{H_5}$ for 16 points, leaving $\boxed{X_8}\,\boxed{A_1}$ in our hand.
   - Play the word $\boxed{A_1}\,\boxed{X_8}$ for 9 points, emptying our hand and ending the game.

   Design and analyze an algorithm to compute the maximum number of points that can be earned from a given sequence of tiles. ***The input consists of two arrays* Letter$[1..n]$*, containing a sequence of letters between A and Z, and* Value$[A..Z]$*, where* Value$[i]$ *is the value of letter i.*** The output is a single number. Assume that you can find all English words that can be made from any seven tiles, along with the point values of those words, in $O(1)$ time.

3. ***Extra credit.*** Submit your answer to Homework 1 problem 4.

CS 473 Fall 2013 — Homework 2 Problem 1

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section: T4 T5 W2 W3 W5 None	

(a) Describe and analyze an algorithm that partitions a list of n integers (positive or negative) into contiguous chunks minimizing the sum of the squares of the sums of the chunks.

(b) Describe and analyze an algorithm that partitions a given list of n integers into ***at most k*** contiguous chunks minimizing the sum of the squares of the sums of the chunks.

CS 473 Fall 2013 — Homework 2 Problem 2

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section: T4 T5 W2 W3 W5 None	

Design and analyze an algorithm that computes the maximum possible score for a given sequence of Scrabble tiles.

CS 473 Fall 2013 — Homework 1 Problem 4

★ ***Extra Credit*** ★

Name:	NetID:
Name:	NetID:
Name:	NetID:
Section: T4 T5 W2 W3 W5 None	

Design and analyze an algorithm to reconstruct an unknown binary tree from the output of Bob's unknown broken tree-traversal algorithms.
