Problem Set 5
Due: Friday, November 8, 2019 (within the first 5 minutes of class).

Problem 1. Activity Selection
Consider a variation of the Activity Selection problem. As in the original problem, we have \( n \) activities to schedule and each activity has a start time \( s_i \) and a finish time \( f_i \), and we cannot schedule two activities if they overlap. However, now each activity has a weight \( w_i \) which is the profit you get for scheduling activity \( a_i \). Our goal is to find a set \( S \) of non-overlapping activities which have maximum total weight. You may assume all the weights are distinct.

(a) \[ 10 \text{ points} \] Consider the following greedy strategy:
(1) sort the activities so \( w_1 > w_2 > \ldots > w_n \).
(2) for \( i = 1 \) to \( n \)
   - Put activity \( a_i \) into \( S \) unless it overlaps an activity already in \( S \).

Give an example (by providing values for \( s_i, f_i, \) and \( w_i \)) where this strategy does not yield an optimal solution.

(b) \[ 15 \text{ points} \] Now suppose each activity has length 1 (so \( f_i = s_i + 1 \)) and the start (and therefore end times) are integers. Formally prove that the algorithm of part (a) finds an optimal solution.

Problem 2. Huffman's Encoding
(a) \[ 6 \text{ points} \] Give a Huffman code for the input string:
ABAACABAACACAACBAACA
Show the tree and the actual binary encoding of the string. If two characters may be assigned the same code, break ties alphabetically, i.e. assign the lower valued code to the character that appears alphabetically first.

(a) \[ 10 \text{ points} \] Let \( k \) denote the number of character types in a file to be encoded (e.g. for the string above, the character types are A, B, and C, so \( k = 3 \)). Suppose the characters are sorted in increasing order of frequency. For this setting, describe how to modify Huffman's algorithm to improve its running time. Provide pseudocode along with your description.

(b) \[ 4 \text{ points} \] Assuming the sort occurs prior to your algorithm, briefly describe the runtime of your algorithm (1-2 sentences is sufficient) in terms of \( k \) (the number of character types).

Problem 3. Divide and Conquer
You are given an array of \( n \) numbers where every value except one appears exactly twice; the duplicate values appear next to each other and the remaining value appears only once.

For example:
8 8 2 2 1 1 4 4 5 5 3 6 6 7 7
3 appears only once.
18 18 23 10 10 19 19 17 17 21 21
23 appears only once.

(a) \[ 10 \text{ points} \] Design an efficient divide and conquer algorithm to find which value appears only once. Briefly describe it in words and provide pseudocode.

(b) \[ 5 \text{ points} \] Express the running time of your algorithm as a recurrence \( T(n) \).

(c) \[ 5 \text{ points} \] Solve your recurrence to big-\( \Theta \) accuracy.