Algorithms (CS:3330:0001 or 22C:031:001)
Homework 5

Each of the following five questions counts for the same number of points. The first problem asks for a greedy algorithm and part of the argument for its correctness. The next two problems are exercises in recursive thinking. The last two problems can be approached by the divide and conquer paradigm.[1]

The homework is due in class on Friday, October 31. If you can't make it to class, drop it in my mailbox in the MacLean Hall mailroom.

1. We are given a set of n jobs – the i-th job has a processing time $t_i > 0$ and a weight $w_i > 0$. We want to order the jobs so as to minimze the weighted sum of completion times, $\sum_{i=1}^{n} w_i C_i$.

   The completion time C_i of job i in a given ordering of the jobs is the sum of the processing times of all jobs up to and including i in the order. For example, for the ordering $\langle 2, 3, 1 \rangle$, the completion time of job 2 is t_2, of job 3 is $t_2 + t_3$, and of job 1 is $t_2 + t_3 + t_1$.

   Design an efficient and correct greedy algorithm for the problem.

   Argue that the greedy algorithm works correctly whenever $n = 2$, that is, for all inputs with two jobs.

2. We are given a $2^n \times 2^n$ chessboard with one (arbitrarily chosen) square removed, where $n \geq 1$ is any integer. Describe a recursive algorithm that tiles such a board without gaps or overlaps using L-shaped pieces, each composed of 3 squares.

   See the figure for example tilings for some inputs with $n = 2$.

   Hint: Think of the $2^n \times 2^n$ chessboard as made up of four $2^{n-1} \times 2^{n-1}$ chessboards.

e	e	d	d
	e	b	d
a	b	b	c
a	a	c	c

a	a	c	c
a	b	b	c
e	b		d
e	e	d	d

Figure 1: Problem 2: Tilings for two instances with $n = 2$. Note that we use 5 L-shaped pieces.

[1] The problems are adapted from the textbook and Jeff Erickson's notes.

3. Consider the following program that is intended to sort its input.

   ```
   StoogeSort(A[0,..,n − 1])
       if n = 2 and A[0] > A[1]
           swap A[0] and A[1]
       else if n > 2
           m = ⌊2n/3⌋
           StoogeSort(A[0,..,m − 1])
           StoogeSort(A[n − m,..,n − 1]
           StoogeSort(A[0,..,m − 1])
   ```

   Does the algorithm always sort its input?[2] What if we replace the line $m = \lfloor 2n/3 \rfloor$ with $m = \lceil 2n/3 \rceil$? Explain.

4. In an array $A[1..n]$ of integers, a pair of numbers $A[i]$ and $A[j]$ form a *significant inversion* if $i < j$ and $A[i] > 3A[j]$. By modifying our algorithm for counting inversions, give an $O(n \log n)$ algorithm for counting the number of significant inversions in a given array.

5. You are at a political convention with n delegates, each one a member of exactly one political party. The number of parties may be greater than 2. It is impossible to tell which political party any delegate belongs to; in particular, you will be summarily ejected if you ask. However, you can determine whether any two delegates belong to the *same* party or not by introducing them to each other – members of the same party always greet each other with smiles and friendly handshakes; members of different parties always greet each other with angry stares and insults.

   Suppose that a majority (strictly more than half) of the delegates are from the same political party. Describe an algorithm that identifies a member (any member) of the majority party using only $O(n \log n)$ introductions.

Let's review some policy from the first day handout. "For homework problems, collaboration is allowed, assuming each of you has first spent some time (about 30 minutes) working on the problem yourself. However, no written transcript (electronic or otherwise) of the collaborative discussion should be taken from the discussion by any participant. Furthermore, discussing ideas is okay but viewing solutions of others is not. It will be assumed that each of you is capable of orally explaining the solution that you turn in, so do not turn in something you don't understand. Students are responsible for understanding this policy; if you have questions, ask for clarification."

Please note that although the correctness of your solution counts for a significant fraction of your score, the quality of your writing and a demonstration that you understood the question and made a serious attempt at solving it also count for a significant fraction.

[2] Though the input to the second recursive call does not begin at index 0 of A, it has a clear meaning: sort the array consisting of the last m elements of A.