MATH 416, PROBLEM SET 3

Comments about homework.

- Solutions to homework should be written clearly, with justification, in complete sentences. Your solution should resemble something you'd write to teach another student in the class how to solve the problem.
- You are encouraged to work with other 416 students on the homework, but solutions must be written independently. Include a list of your collaborators at the top of your homework.
- You should submit your homework on Gradescope, indicating to Gradescope where the various pieces of your solutions are. The easiest (and recommended) way to do this is to start a new page for each problem.
- Attempting and struggling with problems is **critical** to learning mathematics. Do not search for published solutions to problems. I don't have to tell you that doing so constitutes academic dishonesty; it's also a terrible way to get better at math.

  If you get stuck, ask someone else for a hint. Better yet, go for a walk.

Problem 1. By pre-sorting the input array at the beginning, show that the Closest-Pair algorithm we described in class can be improved to have worst-case running time $O(n \log n)$ (where n is the number of points), as advertised.

Problem 2.

(a) For a given constant $\delta > 0$, find an example (with proof) of an increasing function $f\colon \mathbb{N} \to \mathbb{N}$ such that f is $O(n^\delta)$, but f is not $O(n^\gamma)$ for any $\gamma < \delta$, and f is not $\Theta(n^\delta)$. Conclude by commenting on the claim, "any f that is $O(n^{\log_b a})$ is covered by one of the first two cases in the Master Theorem."

Recall the third case of the Master Theorem (assuming $a \geq 1$ and $b > 1$ are constants, $f\colon \mathbb{N} \to \mathbb{N}$ is increasing, ...):

> If both
> (i) f is $\Omega(n^\gamma)$ for some constant $\gamma > \log_b a$, and
> (ii) there is a constant $c < 1$ such that $af(n/b) \leq cf(n)$ for all n sufficiently large,
> then $T(n)$ is $\Theta(f(n))$.

(b) Show that if $f(n) = n^{\log_b a}$, then (ii) is false.

(c) Find an example of a function f for which (i) holds but (ii) fails. (You may choose your favorite values of a and b, e.g., $a = b = 2$.) (*Hint:* One approach is to build a function f for which the inequality $af(n/b) \geq f(n)$ holds for infinitely many n. You can choose which n these are in advance, and then you have a lot of freedom to decide the remaining values of the function in order for (i) to hold.)

(d) Show that in fact (ii) *implies* (i). (For simplicity you may consider only n that are exact powers of b.)

Problem 3. A *partial matching* of $[n] = \{1, \ldots, n\}$ is simply a set of pairwise-disjoint 2-element subsets of $[n]$. That is, it's a way of pairing up some of the elements of $[n]$, possibly leaving some unpaired. For example, $\{\{2,5\},\{3,6\},\{1,7\}\}$ is a partial matching of $[8]$ (in which 4 and 8 are left unpaired). (The pairs are unordered.)

(a) List or draw all partial matchings of $[4]$.

(b) Let m_n be the number of partial matchings of $[n]$. Prove that this sequence satisfies the recurrence $m_{n+1} = m_n + nm_{n-1}$ for $n \geq 1$ and $m_0 = m_1 = 1$.

(c) The *exponential generating function* of a sequence[1] (a_n) is the (formal) power series

$$\sum_{n=0}^{\infty} \frac{a_n}{n!} z^n.$$

Let $M(z)$ be the exponential generating function of (m_n). Verify that $M(z)$ and $e^{z+\frac{1}{2}z^2}$ each satisfy the initial value problem $\frac{d}{dz}M(z) = (1+z)M(z)$, $M(0) = 1$ (and hence are equal).

Problem 4. Suppose that you are given n nonvertical lines in the plane, labeled $L_1, \ldots, L_n$, with the i^{th} line specified by the equation $y = a_i x + b_i$. Assume also that no three lines intersect in a single point. Say that the line L_i is *uppermost* at an x-coordinate x_0 if $a_i x_0 + b_i > a_j x_0 + b_j$ for all $j \neq i$. Say that the line L_i is *visible* if there is some x-coordinate at which it is uppermost. (Intuitively, this means that some portion of the line can be seen "looking down from $y = \infty$.") Give (with proof) an algorithm that takes n lines as input and (with proof) in $O(n \log n)$ time returns exactly the visible lines.

(*Hint:* First, sort the list of lines by slope. Then recursively apply the algorithm to the first $n/2$ lines and to the second $n/2$ lines. But it won't be enough to know which of the first $n/2$ lines are visible and which of the second $n/2$ lines are visible; your algorithm should report a bit more than that. (Consider the case $n = 4$.))

Problem 5. Suppose you are given a $2^n \times 2^n$ checkerboard with one (arbitrarily chosen) square removed. Describe an algorithm in pseudocode that computes a tiling of the board by **L**-shaped tiles, each composed of exactly three squares. Your input is the integer n and two n-bit integers representing the row and column of the missing square. The output is a list of the positions and orientations of $(4^n - 1)/3$ tiles. Your algorithm should run in $O(4^n)$.

[1] The exponential generating function is often useful for counting objects that involve some choice of ordering.