MAI0102 ENUMERATIVE COMBINATORICS, 2018
PROBLEM SET 5

Each problem is worth ten points. You may get partial credit for non-useless, non-perfect solutions. The problems are not ordered by difficulty. Hand in your solutions no later than December 6.

1. (This is closely related to Stanley's Exercise 3.57.) A poset P is a *forest* if $I_x \cap I_y = \emptyset$ whenever $x, y \in P$ are incomparable elements. (Here, we use the notation $I_p = \{q \in P \mid q \leq p\}$ for $p \in P$.) Suppose P is a forest with n elements. Prove that

$$e(P) = \frac{n!}{\prod_{x \in P} |I_x|},$$

where $e(P)$ denotes the number of linear extensions of P.
Hint. A possible approach is to induct on $|P|$. If P does not have $\hat{1}$, P is a disjoint union of two smaller forests. Then use the result of Stanley's Example 3.5.4.

2. Let $d(n)$ denote the dimension of the incidence algebra, i.e. the number of nonempty intervals, of the partition lattice Π_n. Show that the exponential generating function satisfies

$$\sum_{n \geq 0} d(n) \frac{x^n}{n!} = e^{e^{e^x - 1} - 1}.$$

Hint. Stanley's identity (1.94b) is probably useful more than once.

3. (This problem is posed in Stanley, page 264.) Let η be the element of the incidence algebra of a locally finite poset P defined by $\eta(a, b) = 1$ if a is covered by b, and $\eta(a, b) = 0$ otherwise. Prove that $(1 - \eta)^{-1}(x, y)$ is the number of maximal chains in the interval $[x, y]$.

4. A bit string (finite sequence of zeros and ones) is called *aperiodic* if it is not a power of a shorter string (string multiplication being concatenation). Thus, 110101 is aperiodic whereas $110110 = (110)^2$ and $101010 = (10)^3$ are not. Prove that every bit string is a power of a unique aperiodic string. In other words, show that if $u^m = v^n$ for aperiodic bit strings u and v, then $u = v$ (and $m = n$).

5. Let $p(n)$ be the number of aperiodic bit strings on n bits. Show that

$$p(n) = \sum_{d \mid n} \mu(d, n) 2^d,$$

where μ denotes the Möbius function of the divisibility lattice D_n.
You may use the result of the previous problem even if you did not solve it.