Partition Project Notes

Robert Dougherty-Bliss

May 2, 2021

AJ and I began studying the average number of parts over partitions of n with largest part k. The average is exactly

$$\frac{1}{p(n,k)} \sum_{p \in P(n,k)} L(p),$$

where $P(n,k)$ is the set of partitions of n with largest part k, $p(n,k)$ is the *number* of such partitions, and $L(p)$ is the number of parts of p. It makes sense to study the sum on its own terms and introduce the normalizing factor $p(n,k)$ later. To that end, let

$$S(n,k) = \sum_{p \in P(n,k)} L(p).$$

The cases $k = 1$ and $k = 2$ are delightful:

$$\begin{aligned} S(n,1) &= n \\ S(n,2) &= \frac{3n^2}{8} + \frac{(-1)^n - 3}{8} n - \frac{(-1)^n - 1}{16}. \end{aligned}$$

The sequence $S(n,2)$ has a more striking representation in terms of the *pentagonal numbers* q_n:

$$S(n,2) = q_{(-1)^n \lfloor n/2 \rfloor}; \qquad q_n = \frac{n(3n-1)}{2}.$$

We know quite a bit about the sequences $S(n,k)$. In particular, we have very direct proofs of the following theorems. The OEIS does not seem to know about these.

Theorem 1. *The generating function* $S_k(t) = \sum_{n \geqslant 0} S(n,k) t^n$ *equals* $Z_k(t) - Z_{k-1}(t)$, *where*

$$Z_k(t) = \sum_{i=1}^{k} \frac{t^i}{(1-t^i)^2} \prod_{\substack{1 \leqslant j \leqslant k \\ j \neq i}} \frac{1}{1-t^j}.$$

This tells us that $S(n,k)$ is C-finite with respect to n for every fixed k. This resolves at least one conjecture in the OEIS. (See A308265, which is $S(n,3)$.)

Theorem 2. *As* $n \to \infty$,

$$S(n,k) \sim \frac{H_k}{k!^2} n^k,$$

where $H_k = \sum_{j=1}^{k} \frac{1}{j}$ *is the* k*th harmonic number.*

It turns out that

$$p(n,k) \sim \frac{n^{k-1}}{(k-1)!k!},$$

so we get an asymptotic formula for our average almost immediately:

$$\frac{S(n,k)}{p(n,k)} \sim \frac{H_k}{k} n.$$

This checks out:

$$\frac{1}{100}\frac{S(100,3)}{p(100,3)} = \frac{297}{490} \approx 0.6061$$
$$\frac{H_3}{3} = \frac{11}{18} = 0.61111\ldots$$

NEXT STEPS Our technique to prove these theorems was to apply a touch of generatingfunctionology. That inspired a few other sums. For example, we can show that

$$A(n,k) = \sum_{p \in P(n,k)} 2^{L(p)} = \Theta(2^n).$$

For another one, we can write down a rational generating function for the "Schur-like" polynomials

$$T_n(x_1, \ldots, x_k) = \sum_{\substack{n(i) \geqslant 0 \\ n(1)+2n(2)+\cdots+kn(k)=n}} x_1^{n(1)} x_2^{n(2)} \cdots x_k^{n(k)}.$$

This is still thinking **small**. We should try thinking **BIG**. I want to perform an automated search for restrictions (DI, congruences, largest part, etc.) that result in sums where gfun can guess a generating function. That, however, is for another time.