

INVESTIGATIONS REGARDING PARTITIONS AND MULTIPARTITIONS

JOANNA FURNO AND PATRICK WATERS

ADVISOR: DR. HOLLY SWISHER
OREGON STATE UNIVERSITY

ABSTRACT. We consider several topics in the theory of partitions, focusing on questions regarding multipartitions. Results include the formulation of a combinatorial explanation of the crank statistic generalized to multipartitions; the conjecture and proof of several multipartition identities; the calculation of the number of k -partitions of n with crank n , $n-1$, or $n-2$; the construction of a crank-reversing bijection on the partitions of n ; the definition, and characterization of movable multipartitions, the proof of several partition identities for movable multipartitions; the definition of friendly partitions, and a generating function for the number of friendly partitions of n .

1. INTRODUCTION

1.1. Some Notation and Definitions. A *partition* of a nonnegative integer n is a nonincreasing sequence of non-negative integers whose sum is n . For example, the sequence $\{4, 1, 0, 0, 0, \dots\}$ is a partition of 5 because $4 + 1 = 5$ (It is standard to write $4 + 1$ instead of the actual sequence). We represent the zero partition, (whose sequence is all zeros) with the symbol \emptyset . Let $p(n)$ denote the number of partitions of n .

A *k -component multipartition* of a nonnegative integer n is a k -tuple of partitions λ_i such that $|\lambda_1| + |\lambda_2| + \dots + |\lambda_k| = n$. For example, $(4 + 1, 1, 2 + 2 + 1, \emptyset)$ is a 4-partition of 11. Let $P_k(n)$ denote the number of k -component multipartitions of n .

Many methods in the theory of partitions use generating functions. The generating function for a certain type of partitions (or multipartitions) is a power series whose n^{th} coefficient is the number of partitions of the given type, of n . For example, the generating function for $p(n)$ is

$$\sum_{n=0}^{\infty} p(n)q^n = \prod_{n=1}^{\infty} \frac{1}{1 - q^n},$$

and the generating function for $P_k(n)$ is

$$\sum_{n=0}^{\infty} P_k(n)q^n = \prod_{n=1}^{\infty} \frac{1}{(1 - q^n)^k}.$$

The second equation is also the generating function for the number of partitions of a nonnegative integer n where each part is assigned one of k different colors. In this sense, some authors refer to multipartitions as colored partitions.

Date: 9/16/2007.

This work was done during the Summer 2005 REU program in Mathematics at Oregon State University.

Ramanujan was the first mathematician to discover and prove congruences among the values of $p(n)$. The simplest and most famous of these congruences are

$$\begin{aligned} p(5n+4) &\equiv 0 \pmod{5}, \\ p(7n+5) &\equiv 0 \pmod{7}, \\ p(11n+6) &\equiv 0 \pmod{11}. \end{aligned}$$

These three congruences can be explained combinatorially by a statistic called the crank, which was conjectured by Dyson and discovered by Andrews and Garvan in [4]. Let $l(\lambda)$ be the largest part occurring in λ , $\omega(\lambda)$ be the number of ones, and $\mu(\lambda)$ be the number of parts greater than $\omega(\lambda)$. Then the *crank* of a partition λ is

$$c(\lambda) = \begin{cases} l & \text{if } \omega = 0 \\ \mu - \omega & \text{otherwise} \end{cases}.$$

The crank explains the congruences by splitting the partitions into sets of equal size. Let $C(j, m, n)$ be the number of partitions of n congruent to $j \pmod{m}$. Then

$$\begin{aligned} C(j, 5, 5n+4) &= \frac{1}{5}p(5n+4), \\ C(j, 7, 7n+5) &= \frac{1}{7}p(7n+5), \\ C(j, 11, 11n+6) &= \frac{1}{11}p(11n+6). \end{aligned}$$

Let $c(m, n)$ be the number of partitions on n with crank m . Then the generating function for $c(m, n)$ is

$$\sum_{n \geq 0} \sum_{m = -\infty}^{\infty} c(m, n) z^m q^n = zq - q + \prod_{j=0}^{\infty} \frac{1 - q^j}{(1 - zq^j)(1 - z^{-1}q^j)}.$$

A special notation for infinite products is sometimes used:

$$(a_1, a_2, \dots, a_k; q)_{\infty} = \prod_{j=0}^{\infty} (1 - a_1 q^j)(1 - a_2 q^j) \cdots (1 - a_k q^j).$$

In this notation, the crank generating function becomes

$$\sum_{n \geq 0} \sum_{m = -\infty}^{\infty} c(m, n) z^m q^n = zq - q + \frac{(q; q)_{\infty}}{(zq, z^{-1}q; q)_{\infty}}.$$

It will sometimes be convenient to refer to “abstract crank” coefficients $\tilde{c}(m, n)$, defined by

$$\sum_{n=0}^{\infty} \tilde{c}(m, n) z^m q^n = \prod_{j=0}^{\infty} \frac{1 - q^j}{(1 - zq^j)(1 - z^{-1}q^j)}.$$

By squaring the crank generating function, Andrews [2] has extended the idea of the crank to explain a congruences of certain 2-component multipartitions (referred to as bipartitions).

If λ is a partition of n , then the conjugate of λ is $\lambda' = k_1 + k_2 + \cdots + k_m$, where k_i is the number of parts of λ that are greater than or equal to i . A Ferrers diagram can be constructed for λ by letting the number of dots in the i th row equal the i th part. The Ferrers diagram for λ' can then be

constructed by letting the number of dots in the i th row equal the number of dots in the i th column of the Ferrers diagram of λ , so the diagram for λ' is obtained by reflecting the Ferrers diagram of λ across the diagonal.

1.2. Summary of Results. In this report we present our results arising from the investigation of questions regarding cranks and multipartitions. In the first section, a bijection is presented between partitions of n with a crank of m and partitions of n with a crank of $-m$. Next, some identities are presented between different values of $c(m, n)$. Finally, the first section contains a combinatorial interpretation of extending the crank to k -component multipartitions.

In the second section we present a combinatorial function for $P_k(n)$ in terms of $p(n)$. This function is then used to prove some congruences for multipartitions.

The third section contains an extension of conjugation to multipartitions. Let the Ferrers diagrams for each component of the multipartition be stacked on top of each other to create a three-dimensional diagram. This three-dimensional diagram can then be rotated. The multipartitions that have diagrams that yield another multipartition by rotation are called movable multipartitions. The third section also discusses the difficulties encountered in attempting to find a generating function for the number of movable multipartitions of n .

While a generating function was not found for movable multipartitions, a generating function was found for a subset of the movable multipartitions. The multipartitions in this subset are called friendly movable multipartitions (fmmp's) and are explained in the fourth section. This section also discusses certain partition identities involving fmmp's.

The results in sections 2.6-3.2 are due to the first author. The results in sections 2.1-2.5 and 4.1-appendices are due to the second author.

2. THE CRANK

2.1. A self inverse crank reversing bijection. Notice that the conjugation operation is a bijection on the partitions of n that reverses the rank of each partition, and that conjugating a partition twice is the identity operation. We wish to construct an analogue to the conjugation operation for the crank statistic, that is a bijection, f , on the partitions of n , so that

$$\begin{aligned} (1) \quad & f^2(\pi) = \pi \\ (2) \quad & c(f(\pi)) = -c(\pi). \end{aligned}$$

Since we know by the symmetry of the crank generating function that $c_k(m, n) = c_k(-m, n)$, to obtain a self inverse function it will suffice to define $f(\pi)$ when $c(\pi) = 0$, and to define an injection from partitions of n with positive crank to those with negative crank.

2.2. Notation and preliminaries. Define $\mathcal{P}_k(n)$ to be the set of k -partitions of n . If $k = 1$, we will write $\mathcal{P}(n)$. The largest part, ones, and number of parts greater than number of ones functions will be assumed to be taken of the partition π , that is, $l = l(\pi)$, $\omega = \omega(\pi)$, and $\mu = \mu(\pi)$.

Consider two partitions, λ of n_1 and π of n_2 . Define $\lambda \circ \pi$ to be the partition whose parts are all the parts of both λ and π . Clearly $\lambda \circ \pi$ is a partition of $n_1 + n_2$. If the parts of λ and π come from disjoint sets, then $\circ : \mathcal{P}(n) \times \mathcal{P}(n) \rightarrow \mathcal{P}(n)$ is an injection. Furthermore, if A is a subset of

$\mathcal{P}(n) \times \mathcal{P}(n)$ and \circ is an injection when applied to A , then $\circ^{-1} : \circ(A) \longrightarrow A$ is an injection from $\circ(A)$ into $\mathcal{P}(n) \times \mathcal{P}(n)$.

Consider two partitions, λ of n_1 and π of n_2 , as sequences (ending with infinitely many zeros). Define $\lambda + \pi$ to be the termwise sum of the two sequences. That is, $(\lambda + \pi)_i = \lambda_i + \pi_i$. If for each i , either λ_i or π_i comes from a set with only one element, then $+$: $\mathcal{P}(n) \times \mathcal{P}(n) \longrightarrow \mathcal{P}(n)$ is an injection. Furthermore, if A is a subset of $\mathcal{P}(n) \times \mathcal{P}(n)$ and $+$ is an injection when applied to A , then $+^{-1} : +(A) \longrightarrow A$ is an injection from $+(A)$ into $\mathcal{P}(n) \times \mathcal{P}(n)$.

2.3. **Definition of f .** Suppose that $\omega = 0$, then we may write $\pi = (l^{b_1} a_2^{b_2} \dots a_t^{b_t})$, then define

$$f(\pi) = (l^{b_1-1} a_2^{b_2} \dots a_t^{b_t} 1^l).$$

The interesting case is when $\mu > \omega > 0$. In this case $f(\pi)$ is defined using the following Ferrers diagrams. This is of course not a rigorous definition. For the skeptical reader, a formal definition and rigorous proof are presented after our diagram. In any case, this diagram is the motivation behind our proof:

FIGURE 1. $f(\mathbf{P})$ when $\mu > \omega > 0$.

Suppose that $\mu > \omega > 0$, then we may write

$$\pi = \{l, x_2, x_3, \dots, x_\mu > \omega, x_{\mu+1} \leq \omega, \dots, x_j > 1, x_{j+1} = 1, \dots, x_{j+\omega} = 1, 0, \dots\}.$$

Now define $s = \pi_{\omega+1} - (\omega + 1)$ (we know that $s \geq 0$), and the partitions α , β , γ , and δ as follows:

$$\alpha_i = \begin{cases} \omega + 1 & \text{if } i \leq \mu \\ \pi_i & \text{if } i > \mu \end{cases}$$

$$\beta_i = \begin{cases} c & \text{if } i \leq \omega + 1 \\ \pi_i - (\omega + 1) & \text{if } \omega + 1 < i \leq \mu \\ 0 & \text{if } i > \mu \end{cases}$$

$$\gamma_i = \begin{cases} \pi_i & \text{if } i \leq \omega \\ 0 & \text{if } i > \omega \end{cases}$$

$$\delta_i = \begin{cases} \mu + 1 & \text{if } i \leq \omega \\ \alpha_{i+\mu-\omega} & \text{if } \omega < i \leq j + \mu - \omega \\ 1 & \text{if } j + \mu - \omega < i \leq j + 2\mu - \omega \\ 0 & \text{if } j + 2\mu - \omega < i. \end{cases}$$

The task of verifying that α , β , γ , and δ are partitions is left to the reader, who may refer to our diagram for guidance. Define $f(\pi) = (\delta \circ \bar{\beta}) + \gamma$.

In the case that $c(\pi) = 0$, then define $f(\pi) = \pi$.

2.4. Proof that f is the the desired function. If $c(\pi)=0$ or $\omega = 0$, then it is clear that f satisfies properties (1) and (2). In the case $\mu > \omega > 0$, it remains to show that f is function, that the codomain is the subset of $\mathcal{P}(n)$ with negative crank, and that f is an injection.

By considering the definitions of α , β , and γ , we see that $\pi = \alpha + \beta + \gamma$, and that since for each i , only one of $\alpha_i, \beta_i, \gamma_i$ may come from a set with more than one element, the map $\pi \mapsto (\alpha, \beta, \gamma)$ is an injection. It is clear that the map $\alpha \mapsto \delta$ is an injection, thus so is $(\alpha, \beta, \gamma) \mapsto (\delta, \beta, \gamma)$, and that δ is a partition of n . Since β has at most μ parts, the greatest part repeated at least $\omega + 1$ times, $\bar{\beta}$ has no part greater than μ , and no part less than $\omega + 1$. Because δ has no parts of size between $\omega + 1$ and μ , we see that the map $(\delta, \beta) \mapsto \delta \circ \bar{\beta}$ is an injection, and thus so is $(\delta, \beta, \gamma) \mapsto (\delta \circ \bar{\beta}, \gamma)$. Since the first ω parts of $\delta \circ \bar{\beta}$ come from the set $\{\mu + 1\}$, and only the first ω parts of γ may be non-zero, the map $(\delta \circ \bar{\beta}, \gamma) \mapsto (\delta \circ \bar{\beta}) + \gamma$ is an injection. So in this case, f is the composition of injections. By considering the definition of f , we see that $f(\pi)$ has μ ones, and $\omega + 1$ terms greater than μ . Thus $c(f(\pi)) = -c(\pi)$. Also, we see that $f(\pi)$ is a partition of n .

2.5. Calculating the coefficients $c_k(\mathbf{m}, \mathbf{m})$, $c_k(\mathbf{m}, \mathbf{m} + 1)$, and $c_k(\mathbf{m}, \mathbf{m} + 2)$. If we could calculate the coefficients $c_k(m, m + a)$ for arbitrary k, m, a then we would have an explicit formula for $c_k(m, n)$, and thus also for $p_k(n)$. Due to the complexity of the formula for $p(n)$ found by Ramanujan and Hardy, one naturally expects the elementary approach used here to fail, however we do manage to calculate $c_k(m, m + a)$ for the first three values of a .

A multipartition Λ is a multipartition of m with crank m , if and only if each component of Λ has only one part. Each such multipartition is described by an k -tuple of whole numbers, those numbers summing to n . Counting these is equivalent to counting the number of ways to distribute m identical balls into k distinguishable urns. This is one of the elementary problems in combinatorics, and it is well known that there $\langle \begin{smallmatrix} k \\ m \end{smallmatrix} \rangle$ ways to do this, where the angle brackets denote multiset coefficients. Thus,

$$c_k(m, m) = \langle \begin{smallmatrix} k \\ m \end{smallmatrix} \rangle = \binom{k+m-1}{m}.$$

To calculate $c_k(m, m+1)$, we examine the crank generating function using basic combinatorics. Recall the definition of $c_k(m, n)$.

$$\sum c_k(m, n) z^m q^n = \prod_{j=1}^{\infty} \frac{(1-q^j)^k}{(1-zq^j)^k (1-z^{-1}q^j)^k}$$

First we manipulate power series to obtain a more workable form of the crank generating function.

$$\begin{aligned} \frac{(1-q^j)}{(1-zq^j)(1-z^{-1}q^j)} &= (1-q^j)(1+zq^j+z^2q^{2j}+\dots)(1+z^{-1}q^j+z^{-2}q^{2j}\dots) \\ &= (1-zq^j)(1+(z+z^{-1})q^j+(z^2+1+z^{-2})q^{2j}\dots) \\ &= ((1+(z-1+z^{-1})q^j+(z^2-z+1-z^{-1}+z^{-2})q^{2j}\dots) \\ &= \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \\ \sum c_k(m, n) z^m q^n &= \left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k \end{aligned}$$

In this product of power series, we see that in no series does any term have a greater power of z than q . Therefore, to compute a coefficient $c_k(m, m+1)$, we need only consider terms each power series with q -exponent at most one more than z -exponent; that is, terms with $z^n q^n$ or $z^n q^{n+1}$. Let f be a power series, then by $r(f)$ we mean then relevant terms of f to our present calculation. Notice that when $j \geq 3$ no nontrivial terms are relevant, and that when $j = 2$, the only nontrivial relevant term is $1+zq^2$. We therefore have

$$\begin{aligned} r \left(\left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k \right) &= r \left(\left[(1+zq^2) \left(\sum_{l=0}^{\infty} z^l q^l - z^l q^{l+1} \right) \right]^k \right) \\ &= r \left(\left[\sum_{l=0}^{\infty} z^l q^k - z^l q^{l+1} + z^{l+1} q^{l+2} \right]^k \right). \end{aligned}$$

We have discarded the irrelevant terms $-z^{l+1}q^{l+3}$. Observe that the sum telescopes, so

$$r \left(\left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k \right) = \left(-q + \sum_{l=0}^{\infty} z^l q^l \right)^k.$$

To get $z^m q^m + 1$ we must choose one $-q$ term and $k - 1$ terms $z^l q^l$ so that $\sum l = m$. There are k ways to choose the $-q$. Choosing the $z^l q^l$ is equivalent to distributing m identical balls into $k - 1$ distinguishable urns. Thus the coefficients are:

$$c_k(m, m + 1) = -k \langle k-1 \rangle_m = -k \binom{k+m-2}{m}.$$

To calculate the coefficients $c_k(m, m + 2)$, we start again from the equation

$$\sum c_k(m, n) z^m q^n = \left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k.$$

We now must use all terms with q -exponent at most two greater than z -exponent. Therefore our function $r(f)$ keeps more terms this time. We now have

$$\begin{aligned} r \left(\left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k \right) &= r \left((1 + zq^3)(1 + zq^2 - q^2 + z^2 q^4) \left(\sum_{l=0}^{\infty} z^l q^l - z^l q^{l+1} + z^{l-1} q^{l+1} \right) \right)^k \\ &= r \left((1 + zq^2 - q^2 + zq^3 + z^2 q^4) \left(\sum_{l=0}^{\infty} z^l q^l - z^l q^{l+1} + z^{l-1} q^{l+1} \right) \right)^k \\ &= r \left(\left[\sum_{l=0}^{\infty} z^l q^l - z^l q^{l+1} + z^{l-1} q^{l+1} + z^{l+1} q^{l+2} \right. \right. \\ &\quad \left. \left. - z^{l+1} q^{l+3} - z^l q^{l+2} + z^{l+1} q^{l+3} + z^{l+2} q^{l+4} \right]^k \right). \end{aligned}$$

Several of these terms telescope, and we are left with

$$r \left(\left[\prod_{j=1}^{\infty} \sum_{l=0}^{\infty} \sum_{i=-l}^l (-1)^{i+l} z^i q^{jl} \right]^k \right) = \left[-q + z^{-1} q + \left(\sum_{l=0}^{\infty} z^l q^l + z^{l+2} q^{l+4} \right) \right]^k.$$

To get $c_k(m, m + 2)$, we choose one term from each of the k terms. We may either choose two $-q$'s, and the rest $z^l q^l$'s; one $z^{-1} q$, and the rest $z^l q^l$'s; or take $z^l q^l$'s from each term, and then choose one term to receive the $+2, +4$. Thus there are

$$c_k(m, m + 2) = \binom{k}{2} \langle k-2 \rangle_m + k \langle k-1 \rangle_{m+1} + k \langle k \rangle_{m-2}$$

ways altogether. We may rewrite this with binomial coefficients

$$c_k(m, m + 2) = \binom{k}{2} \binom{k+m-3}{m} + k \binom{k+m-1}{m+1} + k \binom{k+m-3}{m-2}.$$

2.6. A combinatorial interpretation of the crank statistic generalized to multipartitions. In [2], Andrews defines $b(m, n)$ by squaring the generating function for the crank

$$(3) \quad \sum_{n \geq 0} \sum_{m=-\infty}^{\infty} b(m, n) z^m q^n = \frac{(q; q)_{\infty}^2}{(zq, z^{-1}q; q)_{\infty}^2}.$$

In a similar way, let us define $c_k(m, n)$ by raising the generating function for the crank to the k^{th} power

$$(4) \quad \sum_{n \geq 0} \sum_{m = -\infty}^{\infty} c_k(m, n) z^m q^n = \frac{(q; q)_{\infty}^k}{(zq, z^{-1}q; q)_{\infty}^k}.$$

Just as Andrews gives an interpretation for $b(m, n)$ in terms of modified bipartitions, an interpretation of $c_k(m, n)$ can be given in terms of modified multipartitions.

First, we will describe the modified partitions and multipartitions. The only partition that is modified is 1. Instead of 1, we use 1_{-1} , 1_0 , or 1_1 . For modified multipartitions, any component in a multipartition that is equal to 1 is replaced with a component equal to 1_{-1} , 1_0 , or 1_1 .

Example 2.1. *The multipartition $(1 + 1, 1, 1)$.*

Instead of $(1 + 1, 1, 1)$, we have the modified multipartitions $(1 + 1, 1_{-1}, 1_{-1})$, $(1 + 1, 1_{-1}, 1_0)$, $(1 + 1, 1_{-1}, 1_1)$, $(1 + 1, 1_0, 1_{-1})$, $(1 + 1, 1_0, 1_0)$, $(1 + 1, 1_0, 1_1)$, $(1 + 1, 1_1, 1_{-1})$, $(1 + 1, 1_1, 1_0)$, and $(1 + 1, 1_1, 1_1)$.

Secondly, we will define the r-crank, a modified version of the crank for partitions, and the k -crank, a version of the crank for k -component multipartitions.

Definition 2.2. *The r-crank of a partition λ is*

$$rc(\lambda) = \begin{cases} c(\lambda) & \text{if } \lambda \neq 1 \\ 1 & \text{if } \lambda = 1_{-1} \\ 0 & \text{if } \lambda = 1_0 \\ -1 & \text{if } \lambda = 1_1. \end{cases}$$

Definition 2.3. *The k -crank of a k -component multipartition $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ is*

$$kc(\Lambda) = \sum_{i=1}^k rc(\lambda_i).$$

These definitions are motivated by the values of the generating function for $n = 1$. There is one partition of 1, which has a crank of -1. However, the generation function (4) gives $[1 + z^1q - z^0q + z^{-1}q + \dots]$, instead of $[1 + z^{-1}q + \dots]$. It is the extra terms for $n = 1$ in the crank generating function that causes the need for extra options for components equal to 1 in a multipartition. The i th component in a k -component multipartition of n corresponds to a term in the i th crank generating function. When these terms are multiplied together, the exponent on the q gives the sum of the components and the exponent on the z gives the sum of the r-cranks of the components.

Thirdly, we will define a weight for k -component multipartitions and use the weight to define $c_k(m, n)$.

Definition 2.4. *If Λ has f_0 components equal to 1_0 , then the weight of a k -component multipartition is*

$$wt(\Lambda) = (-1)^{f_0}.$$

Theorem 2.5. *The coefficients $c_k(m, n)$ represent the sum of the weights of the k -component multipartitions of n with k -crank equal to m .*

Proof. Consider the fact that the coefficient for the z^0q term is -1. Again, the i th component in a k -component multipartition of n corresponds to a term in the i th crank generating function. These terms are multiplied together, so the coefficients are multiplied together. The coefficients for all the terms of the crank generating function are positive, except for the z^0q term. Thus, the sign depends on the number of components equal to 1_0 , which are the component that correspond to the z^0q term. \square

Finally, if $c_k(m, n)$ is going to act as a statistic for multipartitions, it would be desirable to have the property that

$$(5) \quad \sum_{m=-\infty}^{\infty} c_k(m, n) = P_k(n).$$

Before we show that this is true, we will prove a lemma.

Lemma 2.6. *For all $l \geq 1$, there are $(3^l + 1)/2$ sequences of modified 1's that have length l and weight 1, and there are $(3^l - 1)/2$ sequences of modified 1's that have length l and weight -1.*

Proof. There are three sequences of length $l = 1$ of modified 1's. These sequences are (1_{-1}) , (1_0) , and (1_1) , which have weights 1, -1 and 1, respectively. Thus there are $(3^1 + 1)/2 = 2$ sequences of modified 1's that have length 1 and weight 1, and there are $(3^1 - 1)/2 = 1$ sequence of length 1 of modified 1's

Suppose that there are $(3^l + 1)/2$ sequences of modified 1's that have length l and weight 1, and that there are $(3^l - 1)/2$ sequences of modified 1's that have length l and weight -1, for some $l \geq 1$. Placing a 1_{-1} at the beginning of each of the sequences of length l does not change the weight of the sequences and gives $(3^l + 1)/2$ sequences of modified 1's that have length $l + 1$ and weight 1 and $(3^l - 1)/2$ sequences of modified 1's that have length $l + 1$ and weight -1. Placing a 1_0 at the beginning of each of the sequences of length l does change the weight of the sequences and gives $(3^l + 1)/2$ sequences of modified 1's that have length $l + 1$ and weight -1 and $(3^l - 1)/2$ sequences of modified 1's that have length $l + 1$ and weight 1. Placing a 1_1 at the beginning of each of the sequences of length l does not change the weight of the sequences and gives $(3^l + 1)/2$ sequences of modified 1's that have length $l + 1$ and weight 1 and $(3^l - 1)/2$ sequences of modified 1's that have length $l + 1$ and weight -1. Thus, there are

$$\frac{2(3^l + 1)}{2} + \frac{3^l - 1}{2} = \frac{3^{l+1} + 1}{2}$$

sequences of modified 1's that have length $l + 1$ and weight 1, and there are

$$\frac{2(3^l - 1)}{2} + \frac{3^l + 1}{2} = \frac{3^{l+1} - 1}{2}$$

sequences of modified 1's that have length $l + 1$ and weight -1. Since the statement hold for $l + 1$, the proof follows by induction. \square

Theorem 2.7. *We have that*

$$\sum_{m=-\infty}^{\infty} c_k(m, n) = P_k(n).$$

Proof. If the generating function were correct for all n , then this would be the case. For each multipartition that does not contain a component equal to 1, the multipartition has a weight of 1 and the correct number of multipartitions are counted. However, it is necessary that to check that the correct number of multipartitions that contain at least one component equal to 1 are counted. Given a particular multipartition that contains at least one component equal to 1, if the sum of the weights of the k -cranks of the corresponding modified multipartitions is equal to 1, then the correct number of multipartitions are counted. Since the components equal to 1 are the only components that change and affect the weight, it is only necessary the sum of the weights of all sequences of modified 1's that have length l is equal to 1, for all $l \geq 1$. By Lemma 2.6, there are $(3^l + 1)/2$ sequences of modified 1's that have length l and weight 1, and there are $(3^l - 1)/2$ sequences of modified 1's that have length l and weight -1. Thus, the sum of the weights of all sequences of modified 1's that have length l is

$$\frac{3^l + 1}{2} - \frac{3^l - 1}{2} = 1.$$

□

It is also interesting to note that Lemma 2.6 implies that there are

$$\frac{3^l + 1}{2} + \frac{3^l - 1}{2} = 3^l$$

different sequences of modified 1's that have length l .

3. SOME MULTIPARTITION CONGRUENCES

3.1. Combinatorial formula for multipartitions. We begin this section by combinatorially deriving a formula for the number of k -component multipartitions of n , in terms of the partitions of n . Let $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ be a k -component multipartition of n . Let this multipartition correspond to the partition of n such that the partition has parts $|\lambda_i|$ in nonincreasing order. For a partition and multipartition to correspond, they must partition the same integer and the number of parts of the partition must be less than or equal to the number of components of the multipartition.

Example 3.1. Note that $(2, 1, \emptyset)$ is a 3-component multipartition of 3. This multipartition corresponds to the partition $2 + 1$. The other multipartitions of 3 with 3 parts that correspond to the partition $2 + 1$ are $(1 + 1, 1, \emptyset)$, $(2, \emptyset, 1)$, $(1 + 1, \emptyset, 1)$, $(1, 2, \emptyset)$, $(1, 1 + 1, \emptyset)$, $(\emptyset, 2, 1)$, $(\emptyset, 1 + 1, 1)$, $(1, \emptyset, 2)$, $(1, \emptyset, 1 + 1)$, $(\emptyset, 1, 2)$, and $(\emptyset, 1, 1 + 1)$.

We begin deriving the formula by counting the number of multipartitions of n with k components that correspond to a partition with no more than k parts. Define a set of partitions that can correspond to multipartitions of n with k components, $S(n, k) = \{\lambda : \lambda \text{ is a partition of } n \text{ with no more than } k \text{ parts}\}$. If λ is a partition of n , let $N(\lambda)$ be the number of parts of λ , and let $f_i(\lambda)$ be the number of parts of λ equal to i . If $\lambda \notin S(n, k)$, then no multipartitions of n with k components correspond to λ . If $\lambda \in S(n, k)$, then there are $\binom{k}{N(\lambda)}$ ways to choose $N(\lambda)$ components to correspond to the parts of λ , i.e., the non-zero components. Next, the parts of λ must be assigned to correspond to the chosen $N(\lambda)$ components. A component corresponds to a part if the part is equal to the integer that the component partitions. Without loss of generality, assign the parts in increasing order. Thus, begin by choosing the components that correspond to the $f_1(\lambda)$ parts equal

to 1. There are $\binom{N(\lambda)}{f_1(\lambda)}$ ways to choose the components to correspond to the $f_1(\lambda)$ parts equal to 1 and $p(1)^{f_1(\lambda)}$ ways to arrange partitions of 1 in the chosen $f_1(\lambda)$ components. Next, there are $\binom{N(\lambda) - f_1(\lambda)}{f_2(\lambda)}$ ways to choose the components to correspond to the $f_2(\lambda)$ parts equal to 2 and $p(2)^{f_2(\lambda)}$ ways to arrange partitions of 2 in the chosen $f_2(\lambda)$ components. In general, there are $\binom{N(\lambda) - \sum_{i=1}^{a-1} f_i(\lambda)}{f_a(\lambda)}$ ways to choose the components to correspond to the $f_a(\lambda)$ parts equal to a and $p(a)^{f_a(\lambda)}$ ways to arrange partitions of a in the chosen $f_a(\lambda)$ components. Thus, there are

$$\binom{k}{N(\lambda)} \cdot \binom{N(\lambda)}{f_1(\lambda)} \cdot p(1)^{f_1(\lambda)} \cdot \prod_{i=2}^n \binom{N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda)}{f_i(\lambda)} \cdot p(i)^{f_i(\lambda)}$$

k -component multipartitions of n that correspond to λ . Since each k -component multipartition of n corresponds to exactly one partition of n ,

$$(6) \quad P_k(n) = \sum_{\lambda \in S(n,k)} \binom{k}{N(\lambda)} \cdot \binom{N(\lambda)}{f_1(\lambda)} \cdot p(1)^{f_1(\lambda)} \cdot \prod_{i=2}^n \binom{N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda)}{f_i(\lambda)} \cdot p(i)^{f_i(\lambda)}.$$

Example 3.2. We compute $P_2(3)$. For $P_2(3)$, $S(n,k) = (3, 2+1)$. Since $N(3) = 1$, $f_1(3) = 0$, $f_2(3) = 0$, $f_3(3) = 1$, and $p(3) = 3$, there are $2 \cdot 3 = 6$ multipartitions of 3 with 2 components that correspond to $\lambda = 3$. They are $(3, \emptyset)$, $(2+1, \emptyset)$, $(1+1+1, \emptyset)$, $(\emptyset, 3)$, $(\emptyset, 2+1)$, and $(\emptyset, 1+1+1)$. Since $N(2+1) = 2$, $f_1(2+1) = 1$, $f_2(2+1) = 1$, $f_3(2+1) = 0$, $p(1) = 1$, and $p(2) = 2$, there are $2 \cdot 2 = 4$ multipartitions of 3 with 2 components that correspond to $\lambda = 2+1$. They are $(2, 1)$, $(1+1, 1)$, $(1, 2)$, and $(1, 1+1)$. Thus, $P_2(3) = 6 + 4 = 10$.

By expanding the binomials and cancelling terms, the formula for $P_k(n)$ can be significantly simplified.

$$\begin{aligned} P_k(n) &= \sum_{\lambda \in S(n,k)} \binom{k}{N(\lambda)} \cdot \binom{N(\lambda)}{f_1(\lambda)} \cdot p(1)^{f_1(\lambda)} \cdot \prod_{i=2}^n \binom{N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda)}{f_i(\lambda)} \cdot p(i)^{f_i(\lambda)} \\ (7) \quad &= \sum_{\lambda \in S(n,k)} \frac{k!}{(k - N(\lambda))! N(\lambda)!} \cdot \frac{N(\lambda)! \cdot p(1)^{f_1(\lambda)}}{(N(\lambda) - f_1(\lambda))! f_1(\lambda)!} \cdot \prod_{i=2}^n \frac{(N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda))!}{(N(\lambda) - \sum_{j=1}^i f_j(\lambda))! f_i(\lambda)!} \cdot p(i)^{f_i(\lambda)} \\ &= \sum_{\lambda \in S(n,k)} \frac{k!}{(k - N(\lambda))!} \cdot \prod_{i=1}^n \frac{p(i)^{f_i(\lambda)}}{f_i(\lambda)!}. \end{aligned}$$

Thus we have shown the following result.

Theorem 3.3. For all $k \geq 1$ and $n \geq 0$,

$$P_k(n) = \sum_{\lambda \in S(n,k)} \frac{k!}{(k - N(\lambda))!} \cdot \prod_{i=1}^n \frac{p(i)^{f_i(\lambda)}}{f_i(\lambda)!}.$$

3.2. Congruences. There are many congruences that can be proven, using the combinatorial formula derived in the previous section. These are congruences of the forms $P_k(n) \equiv 0 \pmod{k}$ and $P_{k+a}(n) \equiv 0 \pmod{k}$. The proofs are presented for k . However, if $d|k$, then $n \equiv 0 \pmod{d}$, so the proofs also hold for d . Also, if $\gcd(k, m) = 1$, let \bar{m} denote the integer such that $m\bar{m} \equiv 1 \pmod{k}$, i.e., the multiplicative inverse of $m \pmod{k}$.

Proposition 3.4. *If $\gcd(k, n!) = 1$, then $P_k(n) \equiv 0 \pmod{k}$.*

Proof. Recall (7), i.e., that

$$P_k(n) = \sum_{\lambda \in \mathcal{S}(n, k)} \frac{k!}{(k - N(\lambda))! N(\lambda)!} \cdot \frac{N(\lambda)! \cdot p(1)^{f_1(\lambda)}}{(N(\lambda) - f_1(\lambda))! f_1(\lambda)!} \cdot \prod_{i=2}^n \frac{(N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda))! p(i)^{f_i(\lambda)}}{(N(\lambda) - \sum_{j=1}^i f_j(\lambda))! f_i(\lambda)!}.$$

In order to simplify notation throughout the rest of the section, let

$$K(n, k, \lambda) = \frac{N(\lambda)! \cdot p(1)^{f_1(\lambda)}}{(N(\lambda) - f_1(\lambda))! f_1(\lambda)!} \cdot \prod_{i=2}^n \frac{(N(\lambda) - \sum_{j=1}^{i-1} f_j(\lambda))! p(i)^{f_i(\lambda)}}{(N(\lambda) - \sum_{j=1}^i f_j(\lambda))! f_i(\lambda)!},$$

which is an integer, since it is a product of expanded binomials and integers. Note that if $\gcd(k, n!) = 1$, then $n!$ has a multiplicative inverse mod k . Since $1 \leq N(\lambda) \leq n$, factoring out $\frac{k}{n!}$ will still leave integer values within the summation. Since $\gcd(k, n!) = 1$,

$$\begin{aligned} P_k(n) &= \frac{k}{n!} \sum_{\lambda \in \mathcal{S}(n, k)} \frac{(k-1)! n!}{(k - N(\lambda))! N(\lambda)!} \cdot K(n, k, \lambda) \\ &\equiv k\bar{n}! \sum_{\lambda \in \mathcal{S}(n, k)} \frac{(k-1)! n!}{(k - N(\lambda))! N(\lambda)!} \cdot K(n, k, \lambda) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

□

Other congruences can be found by calculating the values of a for which $P_{k+a}(n) \equiv 0 \pmod{k}$, given a value for n and the restrictions that $k + a \geq n$ and that $\gcd(k, n!) = 1$. The previous proposition shows that $a = 0$ is a value for all n and $\gcd(k, n!) = 1$. Let $f(k, a) = P_{k+a}(n)$ for the given value of n and $k + a \geq n$. We can then use the rational root test to find the values of a for which $f(0, a) = 0$.

Proposition 3.5. *Let i be an integer such that $f(0, i) = 0$. Let $a \equiv i \pmod{k}$. If $k + a \geq n$ and $\gcd(k, n!) = 1$, then $P_{k+a}(n) \equiv 0 \pmod{k}$.*

Proof. Note that

$$\begin{aligned} f(0, i) &= \sum_{\lambda \in \mathcal{S}(n, k)} \frac{(a)!}{(a - N(\lambda))! N(\lambda)!} \cdot K(n, k + a, \lambda) \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Since $1 \leq N(\lambda) \leq n$, and we have $\gcd(k, n!) = 1$,

$$\begin{aligned}
 P_{k+a}(n) &= \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a)!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) \\
 &= \frac{k+a}{n!} \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a-1)!n!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) \\
 &= \frac{k}{n!} \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a-1)!n!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) + \\
 &\quad \frac{a}{n!} \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a-1)!n!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) \\
 &= \frac{k}{n!} \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a-1)!n!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) + \\
 &\quad \frac{(k-1+a) \cdots (1+a)}{(k-1+a-N(\lambda)) \cdots (1+a-\lambda)} \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(a)!}{(a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) \\
 &= k\bar{n}! \sum_{\lambda \in \mathcal{S}(n,k)} \frac{(k+a-1)!n!}{(k+a-N(\lambda))!N(\lambda)!} \cdot K(n, k+a, \lambda) + \\
 &\quad \frac{(k-1+a) \cdots (1+a)}{(k-1+a-N(\lambda)) \cdots (1+a-\lambda)} f(0, i) \\
 &\equiv 0 \pmod{k}.
 \end{aligned}$$

□

Example 3.6. We calculate the values of a for which $P_{k+a}(2) \equiv 0 \pmod{k}$. For $k > 0$ and $k+a \geq 2$,

$$\begin{aligned}
 P_{k+a}(2) &= 2 \binom{k+a}{1} + \binom{k+a}{2} \\
 &= 2(k+a) + \frac{(k+a)(k+a-1)}{2} \\
 &= \frac{4k+4a}{2} + \frac{k^2-k+2ak+a^2-a}{2} \\
 &= \frac{k^2 + (2a+3)k + (a^2+3a)}{2}.
 \end{aligned}$$

Since $a^2 + 3a = 0$ for $a = 0$ and -3 , we have the following two congruences. For $k > 0$, $\gcd(k, 2) = 1$, $k+a \geq 2$, and $a \equiv 0 \pmod{k}$,

$$\begin{aligned}
 P_{k+a}(2) &\equiv \frac{k^2+3k}{2} \pmod{k} \\
 &\equiv \bar{2}k(k+3) \pmod{k} \\
 &\equiv 0 \pmod{k}.
 \end{aligned}$$

Note that this also gives us $P_{k+a}(2) \equiv 0 \pmod{k+3}$, for $k > 0$, $\gcd(k,2) = 1$, $k+a \geq 2$, and $a \equiv 0 \pmod{k}$.

For $k > 0$, $\gcd(k,2) = 1$, $k+a \geq 2$, and $a \equiv -3 \pmod{k}$.

$$\begin{aligned} P_{k+a}(2) &\equiv \frac{k^2 - 3k}{2} \pmod{k} \\ &\equiv \bar{2}k(k-3) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(2) \equiv 0 \pmod{k-3}$, for $k-3 > 0$, $\gcd(k,2) = 1$, $k+a \geq 2$, and $a \equiv -3 \pmod{k}$.

Example 3.7. We calculate the a values for which $P_{k+a}(3) \equiv 0 \pmod{k}$. For $k > 0$ and $k+a \geq 3$,

$$\begin{aligned} P_{k+a}(3) &= 3 \binom{k+a}{1} + 2 \binom{k+a}{2} \binom{2}{1} + \binom{k+a}{3} \\ &= 3(k+a) + 2(k+a)(k+a-1) + \frac{(k+a)(k+a-1)(k+a-2)}{3!} \\ &= (3k+3a) + (2k^2+4ak-2k+2a^2-2a) + \frac{(k^2+2ak-k+a^2-a)(k+a-2)}{6} \\ &= \frac{18k+18a}{6} + \frac{12k^2+(24a-12)k+12a^2-12a}{6} + \\ &\quad \frac{k^3+(3a-3)k^2+(3a^2-6a+2)k+a^3-3a^2+2a}{6} \\ &= \frac{k^3+(3a+9)k^2+(3a^2+18a+8)k+(a^3+9a^2+8a)}{6}. \end{aligned}$$

Since $a^3 - 9a^2 + 8a = 0$ for $a = 0, -1, \text{ and } -8$, we have the following three congruences. For $k > 0$, $\gcd(k,3) = 1$, $k+a \geq 3$, and $a \equiv 0 \pmod{k}$,

$$\begin{aligned} P_{k+a}(3) &\equiv \frac{k^3+9k^2+8k}{6} \pmod{k} \\ &\equiv \bar{6}k(k+1)(k+8) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(3) \equiv 0 \pmod{k+1}$ and $P_{k+a}(3) \equiv 0 \pmod{k+8}$, for $k > 0$, $\gcd(k,3) = 1$, $k+a \geq 3$, and $a \equiv 0 \pmod{k}$.

For $k > 0$, $\gcd(k,3) = 1$, $k+a \geq 3$, and $a \equiv -1 \pmod{k}$,

$$\begin{aligned} P_{k+a}(3) &\equiv \frac{k^3+6k^2-7k}{6} \pmod{k} \\ &\equiv \bar{6}k(k-1)(k+7) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(3) \equiv 0 \pmod{k-1}$ ($k-1 > 0$) and $P_{k+a}(3) \equiv 0 \pmod{k+7}$ ($k+7 > 0$), for $\gcd(k,3) = 1$, $k+a \geq 3$, and $a \equiv -1 \pmod{k}$.

For $k > 0$, $\gcd(k, 3) = 1$, $k + a \geq 3$, and $a \equiv -8 \pmod{k}$,

$$\begin{aligned} P_{k+a}(3) &\equiv \frac{k^3 - 15k^2 + 56k}{6} \pmod{k} \\ &\equiv \bar{6}k(k-7)(k-8) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(3) \equiv 0 \pmod{k-7}$ ($k-7 > 0$) and $P_{k+a}(3) \equiv 0 \pmod{k-8}$ ($k-8 > 0$), for $k > 0$, $\gcd(k, 3) = 1$, $k + a \geq 3$, and $a \equiv -8 \pmod{k}$.

Example 3.8. We calculate the values of a for which $P_{k+a}(4) \equiv 0 \pmod{k}$. For $k > 0$ and $k + a \geq 4$,

$$\begin{aligned} P_{k+a}(4) &= 5 \binom{k+a}{1} + 3 \binom{k+a}{2} \binom{2}{1} + 2^2 \binom{k+a}{2} + 2 \binom{k+a}{3} \binom{3}{1} + \binom{k+a}{4} \\ &= 5(k+a) + 5(k+a)(k+a-1) + (k+a)(k+a-1)(k+a-2) + \\ &\quad \frac{(k+a)(k+a-1)(k+a-2)(k+a-3)}{24} \\ &= \frac{k^4 + (4a+18)k^3 + (6a^2+54a+59)k^2 + (4a^3+54a^2+118a+42)k}{24} + \\ &\quad \frac{(a^4+18a^3+59a^2+42a)}{24}. \end{aligned}$$

Since $a^4 + 18a^3 + 59a^2 + 42a = 0$ for $a = 0, -1, -3$, and -14 , we have the following four congruences.

For $k > 0$, $\gcd(k, 24) = 1$, $k + a \geq 4$, and $a \equiv 0 \pmod{k}$,

$$\begin{aligned} P_{k+a}(4) &\equiv \frac{k^4 + 18k^3 + 59k^2 + 42k}{24} \pmod{k} \\ &\equiv \bar{24}k(k+1)(k+3)(k+14) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(4) \equiv 0 \pmod{k+1}$, $P_{k+a}(4) \equiv 0 \pmod{k+3}$, and $P_{k+a}(4) \equiv 0 \pmod{k+14}$, for $k > 0$, $\gcd(k, 4) = 1$, $k + a \geq 4$, and $a \equiv 0 \pmod{k}$.

For $k > 0$, $\gcd(k, 24) = 1$, $k + a \geq 4$, and $a \equiv -1 \pmod{k}$,

$$\begin{aligned} P_{k+a}(4) &\equiv \frac{k^4 + 14k^3 + 11k^2 - 26k}{24} \pmod{k} \\ &\equiv \bar{24}k(k-1)(k+2)(k+13) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(4) \equiv 0 \pmod{k-1}$ ($k-1 > 0$), $P_{k+a}(4) \equiv 0 \pmod{k+2}$, and $P_{k+a}(4) \equiv 0 \pmod{k+13}$, for $k > 0$, $\gcd(k, 4) = 1$, $k + a \geq 4$, and $a \equiv -1 \pmod{k}$.

For $k > 0$, $\gcd(k, 24) = 1$, $k + a \geq 4$, and $a \equiv -14 \pmod{k}$,

$$\begin{aligned} P_{k+a}(4) &\equiv \frac{k^4 - 38k^3 + 479k^2 - 2002k}{24} \pmod{k} \\ &\equiv \overline{24}k(k-11)(k-13)(k-14) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(4) \equiv 0 \pmod{k-11}$ ($k-11 > 0$), $P_{k+a}(4) \equiv 0 \pmod{k-13}$ ($k-13 > 0$), and $P_{k+a}(4) \equiv 0 \pmod{k-14}$ ($k-14 > 0$), for $\gcd(k, 4) = 1$, $k + a \geq 4$, and $a \equiv -14 \pmod{k}$.

Example 3.9. We calculate the values of a for which $P_{k+a}(5) \equiv 0 \pmod{k}$. For $k > 0$ and $k + a \geq 5$,

$$\begin{aligned} P_{k+a}(5) &= 7 \binom{k+a}{1} + 5 \binom{k+a}{2} \binom{2}{1} + 3 \cdot 2 \binom{k+a}{2} \binom{2}{1} + 3 \binom{k+a}{3} \binom{3}{1} + \\ &\quad 2^2 \binom{k+a}{3} \binom{3}{1} + 2 \binom{k+a}{4} \binom{4}{1} + \binom{k+a}{5} \\ &= 7(k+a) + 11(k+a)(k+a-1) + \frac{7(k+a)(k+a-1)(k+a-2)}{2} + \\ &\quad \frac{(k+a)(k+a-1)(k+a-2)(k+a-3)}{3} + \\ &\quad \frac{(k+a)(k+a-1)(k+a-2)(k+a-3)(k+a-4)}{120} \\ &= \frac{k^5 + (5a+30)k^4 + (10a^2+120a+215)k^3 + (10a^3+180a^2+645a+450)k^2}{120} + \\ &\quad \frac{(5a^4+120a^3+645a^2+900a+144)k + (a^5+30a^4+215a^3+450a^2+144a)}{120}. \end{aligned}$$

Since $a^5 + 30a^4 + 215a^3 + 450a^2 + 144a = 0$ for $a = 0, -3$, and -6 , we have the following three congruences.

For $k > 0$, $\gcd(k, 120) = 1$, $k + a \geq 5$, and $a \equiv 0 \pmod{k}$,

$$\begin{aligned} P_{k+a}(5) &\equiv \frac{k^5 + 30k^4 + 215k^3 + 450k^2 + 144k}{120} \pmod{k} \\ &\equiv \overline{120}k(k+3)(k+6)(k^2+21k+8) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(5) \equiv 0 \pmod{k+3}$ and $P_{k+a}(5) \equiv 0 \pmod{k+6}$, for $k > 0$, $\gcd(k, 5) = 1$, $k + a \geq 5$, and $a \equiv 0 \pmod{k}$.

For $k > 0$, $\gcd(k, 120) = 1$, $k + a \geq 5$, and $a \equiv -3 \pmod{k}$,

$$\begin{aligned} P_{k+a}(5) &\equiv \frac{k^5 + 15k^4 - 55k^3 - 135k^2 + 414k}{120} \pmod{k} \\ &\equiv \overline{120}k(k-3)(k+3)(k^2+15k-46) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(5) \equiv 0 \pmod{k-3}$ ($k-3 > 0$) and $P_{k+a}(5) \equiv 0 \pmod{k+3}$, for $k > 0$, $\gcd(k, 5) = 1$, $k+a \geq 5$, and $a \equiv -3 \pmod{k}$.

For $k > 0$, $\gcd(k, 120) = 1$, $k+a \geq 5$, and $a \equiv -6 \pmod{k}$,

$$\begin{aligned} P_{k+a}(5) &\equiv \frac{k^5 - 145k^3 + 900k^2 - 1476k}{120} \pmod{k} \\ &\equiv \overline{120}k(k-3)(k-6)(k^2+9k-82) \pmod{k} \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Note that this also gives us $P_{k+a}(5) \equiv 0 \pmod{k-3}$ ($k-3 > 0$) and $P_{k+a}(5) \equiv 0 \pmod{k-6}$ ($k-6 > 0$), for $\gcd(k, 5) = 1$, $k+a \geq 5$, and $a \equiv -3 \pmod{k}$.

We have two more congruences to present. Unlike the previous congruences, these do not seem to generalize to $n > 3$.

Let $a = 3^i j$, $i > 0$, $j > 0$, and $k = 3a = 3^{i+1} j$. Then $k+a = 3 \cdot 3^i j + 3^i j = 4 \cdot 3^i j$, and

$$\begin{aligned} P_{k+a}(2) &= \frac{k^2 + (2a+3)k + (a^2+3a)}{2} \\ &= \frac{3^{2i+2} j^2 + (2 \cdot 3^i j + 3)3^{i+1} j + (3^{2i} j^2 + 3^{i+1} j)}{2} \\ &= \frac{3^{i+1} j(3^{i+1} j + 2 \cdot 3^i j + 3 + 3^{i-1} j + 1)}{2} \\ &= \frac{3^{i+1} j(16 \cdot 3^{i-1} j + 4)}{2} \\ &= k(8 \cdot 3^{i-1} j + 2) \\ &\equiv 0 \pmod{k}. \end{aligned}$$

Let $a = 2^i j$, $i > 1$, $j > 0$, and $k = 2a = 2^{i+1} j$. Then $k+a = 2 \cdot 2^i j + 2^i j = 3 \cdot 2^i j$, and

$$\begin{aligned} P_{k+a}(3) &= \frac{k^3 + (3a+9)k^2 + (3a^2+18a+8)k + (a^3+9a^2+8a)}{6} \\ &= \frac{2^{3i+3} j^3 + (3 \cdot 2^i j + 9)2^{2i+2} j^2 + (3 \cdot 2^{2i} j^2 + 9 \cdot 2^{i+1} j + 8)2^{i+1} j}{6} + \\ &\quad \frac{(2^{3i} j^3 + 9 \cdot 2^{2i} j^2 + 2^{i+3} j)}{6} \\ &= \frac{2^{i+1} j(15 \cdot 2^{2i-1} j^2 + 12 \cdot 2^{2i-1} j + 81 \cdot 2^{i-1} j + 12)}{6} \\ &= k(2^{2i-2} 5j^2 + 2^{2i} j + 27 \cdot 2^{i-2} j + 2) \\ &\equiv 0 \pmod{k}. \end{aligned}$$

4. MOVABLE MULTIPARTITIONS

One way to visualize a multipartition is to draw the Ferrers graph of each component and then imagine a three dimensional lattice of dots where the $i+1^{\text{st}}$ Ferrers graph is stacked on top of the

i^{th} , with the first dot of the first part of each component aligned. We may label the position of each dot with coordinate axes, so define the x -axis to point in the direction so that increasing x moves to the next dot, of the same part, of the same component; the y -axis to point in the direction so that increasing y moves to the same dot, of the next part, of the same component; and the z -axis to point in the direction so that increasing z moves to the same dot, of the same part, of the next component. Let the coordinate axes intersect at the first dot of the first part of the first component.

When considering the Ferrers graph of a single partition, we may obtain the partition's conjugate by permuting the x and y axes. Such an operation allows us to make partition identities. For example, the number of partitions of n with parts of size at most m equals the number of partitions of n with at most m parts. We wish to make similar identities for multipartitions by permuting the x , y , and z axes, but if we perform such an operation, we may no longer have a multipartition. The conditions that we require for the image of a multipartition Λ to be a multipartition are that its parts are well defined, and that its components are partitions. For example, if $\Lambda = (1, 0, 1)$, and we permute the x and z axes, then we get a dot, then a space, then a dot on the x -axis, so the first part of the first component is not well defined. Also, it may happen that some component of some image may not be a nonincreasing sequence. We cannot allow this.

We denote the j^{th} part of the i^{th} component of the multipartition Λ by Λ_{ij} . We will consider permutations of the x , y , and z axes as elements of the symmetric group on three elements, $S_{\{x,y,z\}}$, and write them in cycle notation. The image of the operation of permuting the axes of Λ by τ will be denoted by $\tau \circ \Lambda$. For example, to refer to the j^{th} part of the i^{th} component of the image of Λ by conjugation of each part, we write $((xy) \circ \Lambda)_{ij}$. Clearly, this action of $S_{\{x,y,z\}}$ on all 3-d lattices of dots (not necessarily multipartitions) is a group action. We seek the largest subset of dot lattices that are multipartitions, and closed under this action.

Consider the part $((xz) \circ \Lambda)_{ij}$. If it is well defined, then it is the number of components of Λ whose i^{th} part is at least j . Thus for this part to be well defined, we require that

$$\Lambda_{ki} \geq j \quad \text{implies} \quad \Lambda_{k+1,i} \geq j.$$

If we require that all parts are well defined, this is equivalent to the condition that

$$\Lambda_{ij} \geq \Lambda_{i+1,j}$$

for all i and j . If we require that the components of $(xz) \circ \Lambda$ are partitions, then the number of components of Λ whose i th part is at least j must be at least the number of components of Λ whose i^{th} part is at least $j+1$. This condition is obvious for all multipartitions, so $\Lambda_{ij} \geq \Lambda_{i+1,j}$ is a necessary and sufficient condition for $(xz) \circ \Lambda_{ij}$ to be a multipartition

Since every partition has a conjugate, $(xy) \circ \Lambda$ is defined for any multipartition Λ . Thus since $(xy)(xz) \circ \Lambda = (xzy) \circ \Lambda$, we have that $\Lambda_{ij} \geq \Lambda_{i+1,j}$ is a necessary and sufficient condition for $(xzy) \circ \Lambda_{ij}$ to be a multipartition as well.

Consider the permutation $(yz) \circ \Lambda$. The permutation (yz) fixes the x -axis, so the parts of $(yz) \circ \Lambda$ are well defined because they are parts of (possibly different components of) Λ . We see that $((yz) \circ \Lambda)_{ij} = \Lambda_{ji}$, so since $\Lambda_{ij} \geq \Lambda_{i+1,j}$, we have that $(yz) \circ \Lambda$ is a multipartition if and only if $\Lambda_{ij} \geq \Lambda_{i+1,j}$. Since $(xy)(yz) \circ \Lambda = (xyz) \circ \Lambda$, we conclude that

$$\Lambda_{ij} \geq \Lambda_{i+1,j}$$

is a necessary and sufficient condition for $\tau \circ \Lambda$ to be a multipartition for any permutation τ . In this case we say that Λ is a *movable multipartition*, or *mmp*.

Because $\Lambda_{i_j} \geq \Lambda_{i_{j+1}}$ and $\Lambda_{i_j} \geq \Lambda_{i+1_j}$, the furthest extent of dots on the z, y , and x axes of an mmp are respectively the number of components, greatest number of parts in any component, and largest part in any component. Thus let $N(k, l, m; n)$ be the number of mmp's of n with at most k components, no component having more than l parts, and no component having any part larger than m . Then the maps $\Lambda \mapsto \tau \circ \Lambda$ are explicit bijections showing that $N(k, l, m; n)$ is a symmetric function with respect to the parameters k, l , and m . Furthermore, we see that for any other identity for which conjugation provides a bijection between two sets of partitions (for example the partitions of n into distinct parts are equinumerous to the partitions of n with at least one part of each number less than the largest part), there is an analogue for mmp's proved by bijections $\Lambda \mapsto \tau \circ \Lambda$.

5. FRIENDLY MOVABLE MULTIPARTITIONS

5.1. A generating function for friendly movable multipartitions. Finding a generating function for the movable multipartitions of n is very hard, and I will give a dollar to the first person to show me a solution. However, if we restrict our attention to 'friendly' multipartitions (fmmp's), whose $i+1^{\text{st}}$ component is the i^{th} component 'plus' some partition (the addition I refer to is to consider the partitions as sequences and add the two sequences term by term), we may find a generating function. To properly discuss fmmps, we introduce an algebraic structure that generalizes the whole numbers, partitions, and fmmp's.

Definition 5.1. Let $M = (S, +, \leq, | \cdot |)$. Then M is a special monoid iff:

1. $+$ is an associative, commutative binary operation on S , with an identity element 0 .
2. \leq is a partial ordering on S .
3. For all $a, b \in S$, $a \leq a + b$
4. $a \leq b$ implies that there exists a unique c such that $a + c = b$.
5. $| \cdot |: S \rightarrow \mathbb{N}^0$ satisfies the the property $|a + b| = |a| + |b|$.

By \mathbb{N}^0 , I mean the whole numbers, $\mathbb{N} \cup \{0\}$. If, for two elements a, b there is a unique c such that $a + c = b$, then we will sometimes write $c = b - a$ (even though a may not have an inverse). We now define a construction that takes one special monoid to another containing it.

Definition 5.2. If M is a special monoid, then define $O(M)$ to be the set

$$\{\mu \in M[x] \mid \mu_{i+1} \leq \mu_i\}$$

with $+$ being the usual addition of polynomials for in $M[x]$. We say that $\mu^1 \leq \mu^2$ if and only if there exists μ^3 such that $\mu^1 + \mu^3 = \mu^2$ (when μ^3 exists, it must be unique). We also define $|\mu| = \sum_i |\mu_i|$.

Clearly, for any special monoid M , we have that $O(M)$ is also a special monoid. Also, we see that $O(M)$ contains M in the same sense that $M[x]$ contains M .

Example 5.3. The whole numbers \mathbb{N}^0 are a special monoid, with $| \cdot |$ defined by $|n| = n$. Let P be the set of partitions, then $P = O(\mathbb{N}^0)$. Let F be the set of fmmp's, then $F = O(P) = O^2(\mathbb{N}^0)$.

Definition 5.4. If M is a special monoid, define $N(M; n)$ to be the number of elements $\mu \in M$ with $|\mu| = n$. Observe that such a number may not exist, however we will only consider cases where $N(M; n)$ exists for all whole numbers n .

We will need the following lemma to defend the way we index our sums in the coming theorem.

Lemma 5.5. *If $N(M; n)$ exists for all whole numbers n , then $N(O(M); n)$ also exists for all n . In this case both M and $O(M)$ are countable.*

Proof. We know that $|\mu| = \sum_i |\mu_i|$. Thus for any element μ of $O(M)$ that contributes $N(O(M); n)$, the numbers $|\mu_i|$ must partition n . Therefore,

$$N(O(M); n) = \sum_{\text{Partitions } \lambda \text{ of } n} \prod_{\text{Parts } m \text{ of } \lambda} N(M; m).$$

The first part of our lemma follows because the partitions of n are finite, every partition has finitely many parts, and $N(M; m)$ is always finite. Let M be a special monoid with $N(M, n)$ always finite, then for each n , index the set $\{\mu \in M \mid |\mu| = n\}$ with natural numbers $m(\mu)$. Map each μ to the prime power $p_n^{m(\mu)}$. This is an injection from M to \mathbb{N} . \square

Here we prove the existence of a function that we will need in the proof of our theorem.

Lemma 5.6. *There is a bijection φ from $O(M)$ to $M[x]$ with $|\mu| = \sum_i i|\varphi(\mu)_i|$.*

Proof. Let $\mu \in O(M)$, then define $\varphi(\mu) = \sum_i (\mu_i - \mu_{i+1})x^i$. Then

$$\begin{aligned} \sum_i i|\varphi(\mu)_i| &= \sum_i i|\mu_i - \mu_{i+1}| \\ &= \sum_{i=0}^d i|\mu_i| - \sum_{i=0}^d i|\mu_{i+1}| \\ &= \sum_{i=0}^d i|\mu_i| - \sum_{i=1}^{d+1} (i-1)|\mu_{i+1}| \\ &= \sum_{i=0}^d |\mu_i| \\ &= |\mu| \end{aligned}$$

In our sum, d was the degree of μ , so when we telescoped the sum, we used that $\mu_{d+1} = 0$. By definition of $O(M)$, $\mu_i - \mu_{i+1}$ exists uniquely, so our function is well defined. It is clear that elements in $O(M)$ are in bijection with their ‘difference sequences’ in $M[x]$. \square

Theorem 5.7. *If $N(M; n)$ exists for all n , then*

$$\sum_{n=0}^{\infty} N(O(M); n)q^n = \prod_{j=0}^{\infty} \sum_{n=0}^{\infty} N(M; n)q^{jn}$$

Proof. By our lemma, $N(M; n)$ exists and $O(M)$ is countable, so index all elements $\mu \in O(M)$ with ‘upstairs’ indices in \mathbb{N} . Then

$$\begin{aligned} \sum_{n=0}^{\infty} N(O(M); n)q^n &= \sum_i q^{|\mu^i|} \\ &= \sum_i q^{\sum_j j|\varphi(\mu^i)_j|}. \end{aligned}$$

Since φ is a bijection, we may instead sum over all elements $\sigma \in M[x]$, again indexed upstairs with natural numbers.

$$\begin{aligned} \sum_{n=0}^{\infty} N(O(M); n)q^n &= \sum_i q^{\sum_j j|\sigma_j^i|} \\ &= \sum_i q^{|\sigma_1^i|} (q^2)^{|\sigma_2^i|} (q^3)^{|\sigma_3^i|} \dots (q^d)^{|\sigma_d^i|} \\ &= \prod_{j=1}^{\infty} \sum_i (q^j)^{|m_i^j|} \end{aligned}$$

In the previous line we have indexed the elements of m , of M with upstairs indices in \mathbb{N} . This step is justified by the following bijection between terms on the last two lines:

$$q^{|\sigma_1^i|} (q^2)^{|\sigma_2^i|} \dots (q^d)^{|\sigma_d^i|} \mapsto q^{|m^{i1}|} (q^2)^{|m^{i2}|} \dots (q^d)^{|m^{id}|}$$

where m^{ij} is the element of M equal to σ_j^i . This is a bijection because each element in $M[x]$ is a finite sequence in M . Notice that each term is mapped to one with the same power of q . Thus we conclude

$$\sum_{n=0}^{\infty} N(O(M); n)q^n = \prod_{j=0}^{\infty} \sum_{n=0}^{\infty} N(M; n)q^{jn}.$$

□

We demonstrate the utility of our theorem in the following corollaries.

Corollary 5.8. *Theorem 5.7 provides us with a round about way to the generating function for $p(n)$. We have that*

$$\sum_{n=0}^{\infty} p(n)q^n = \prod_{j=1}^{\infty} \frac{1}{1 - q^j}.$$

Proof. We proceed by writing $p(n)$ in terms of the O operator and special monoids, and directly apply our theorem.

$$\begin{aligned} \sum_{n=0}^{\infty} p(n)q^n &= \sum_{n=0}^{\infty} N(O(\mathbb{N}^0); n)q^n \\ &= \prod_{j=1}^{\infty} \sum_{n=0}^{\infty} N(\mathbb{N}^0; n)q^{jn} \\ &= \prod_{j=1}^{\infty} \sum_{n=0}^{\infty} q^{jn} \\ &= \prod_{j=1}^{\infty} \frac{1}{1 - q^j}. \end{aligned}$$

□

We now obtain our generating function for the number of fmmp's of n .

Corollary 5.9. *Let $d(j)$ be the number of divisors of j , then*

$$\sum_{n=0}^{\infty} fmmp(n)q^n = \prod_{j=1}^{\infty} \frac{1}{(1-q^j)^{d(j)}}.$$

Proof. Again, we directly apply Theorem 5.7.

$$\begin{aligned} \sum_{n=0}^{\infty} fmmp(n)q^n &= \sum_{n=0}^{\infty} N(\mathcal{O}^2(\mathbb{N}^0; n)q^n) \\ &= \prod_{j=1}^{\infty} \sum_{n=0}^{\infty} N(\mathcal{O}(\mathbb{N}^0; n)q^{jn}) \\ &= \prod_{j=1}^{\infty} \sum_{n=0}^{\infty} p(n)q^{jn} \\ &= \prod_{j=1}^{\infty} \prod_{i=1}^{\infty} \frac{1}{1-q^{ij}} \\ &= \prod_{j=1}^{\infty} \frac{1}{(1-q^j)^{d(j)}}. \end{aligned}$$

The final step is verified by writing the terms of the double product in a table and observing that $\frac{1}{1-q^n}$ appears once in each row for which the first entry is $\frac{1}{1-q^k}$ with k a divisor of n . \square

Corollary 5.10. *We have the following generalization of our result to n -dimensional arrays of dots satisfying a generalized friendly condition:*

$$\sum_{n=0}^{\infty} N(\mathcal{O}^k(\mathbb{N}^0); n)q^n = \prod_{j=0}^{\infty} \frac{1}{(1-q^j)^{\sum_{j_1|j} \sum_{j_2|j_1} \sum_{j_3|j_2} \cdots \sum_{j_{k-1}|j_k} 1}}.$$

Proof. By induction, the inductive step using our theorem, we see that

$$\sum_{n=0}^{\infty} N(\mathcal{O}^k(\mathbb{N}^0); n)q^n = \prod_{j_1=1}^{\infty} \prod_{j_2=1}^{\infty} \cdots \prod_{j_k=1}^{\infty} \frac{1}{1-q^{j_1 j_2 \cdots j_k}}.$$

Also by induction, the inductive step using the reasoning from the end of the last proof, we get the result. \square

5.2. A partition identity involving fmmp's. I present here a partition identity motivated by the generating function for $fmmp(n)$. Recall that

$$\sum_{n=0}^{\infty} fmmp(n)q^n = \prod_{j=1}^{\infty} \frac{1}{(1-q^j)^{d(j)}}.$$

If j were always prime, or a product of some known number of primes, then we could compute $d(j)$, and the result would be the generating function for some subset of multipartitions having a fixed number of components.

Let $A(n)$ be the number of fmmp's Λ of n who's i^{th} and $i+1^{st}$ components are equal except possibly when i is prime, when the difference between Λ_i and Λ_{i+1} is a partition with only prime parts. An example of such a partition would be $(10+5, 10+5, 7+3, 5+3, 5+3)$. Let $B(n)$ be the number of tri-partitions of n whose first and second components have only parts that are products

of exactly two primes, and whose third component has only parts that are squares of primes. An example of such a partition would be $(15 + 10, 21 + 6, 25 + 16 + 4)$. Then we have the following partition identity.

Theorem 5.11. *For all n , $A(n) = B(n)$.*

Proof. The details of this proof are analogous to the details of the proof of our theorem in section 4. Specifically the same bijection ϕ is used. By p_i we mean the i^{th} prime. By $P[x]$ we mean the polynomials over x whose coefficients are partitions with only prime parts, and by $pr(j)$ we mean the number of partitions of j into prime parts. We have

$$\begin{aligned}
 \sum_{n=0}^{\infty} A(n)q^n &= \sum_{\Lambda \in \mathcal{S}} q^{|\Lambda|} \\
 &= \sum_{\Lambda \in \mathcal{S}} q^{\sum_i p_i |\phi(\Lambda)_{p_i}|} \\
 &= \sum_{a \in P[x]} q^{\sum_i p_i |a_{p_i}|} \\
 &= \prod_{i=1}^{\infty} \sum_{\pi \in P} (q^{p_i})^{|\pi|} \\
 &= \prod_{i=1}^{\infty} \sum_{j=0}^{\infty} pr(j)(q^{p_i})^j \\
 &= \prod_{i=1}^{\infty} \prod_{j=1}^{\infty} \frac{1}{1 - q^{p_i p_j}} \\
 &= \prod_{\substack{\text{products } n \\ \text{of two primes}}} \frac{1}{(1 - q^n)^{d(n)}} \\
 &= \prod_{\substack{\text{products } n \text{ of two} \\ \text{distinct primes}}} \frac{1}{(1 - q^n)^2} \prod_{\substack{\text{squares } n \\ \text{of primes}}} \frac{1}{1 - q^n} \\
 &= \sum_{m=0}^{\infty} B(m)q^m.
 \end{aligned}$$

□

More identities like this could easily be made for friendly partitions whose parts have more complicated prime factorizations, and whose difference sequences have nonzero i^{th} terms where i has a more complicated prime factorization.

6. APPENDIX 1: PROGRAMS

We have written programs in C++ and maple that can be used to search for numerical evidence of partition congruences.

6.1. Program mpartition2. The program mpartition2, written in C++, has global variables n , m , and p . It displays a table showing the abstract crank coefficients $c_m(a, b)$ for b between 0 and n , and a between $-n$ and n . The parameter b changes along the vertical axis, and the parameter a changes on the horizontal axis. Then for each positive number c such that $c = n - dp$ it displays the “number” of m -partitions of c with abstract crank in each residue class (mod p). An example of the use of this program would be to numerically check the first five cases of Andrews’ theorem that for $n \geq 1$, $C_2(5n + 3, j, 5)$ are equal for each j (this implies the congruence $p_2(5n + 3) \equiv 0 \pmod{5}$). To do this, one would set the global variables $n = 28$, $m = 2$, and $p = 5$, and then compile and run the program. By changing the parameters, n , m , and p , one may check whether the crank stastic explains a conjectured partition congruence.

```
#include <iostream>
using namespace std;

int n = 19;
int m = 1;
int p = 5;

void printQZSeries(int* f)
{
  for (int x = 0; x <= n; x++){
    for (int y = 0; y <= 2*n; y++){
      cout << f[(2*n+1)*x + y] << " ";
    }
    cout << "\n";
  }
};

int* multiply2(int* f, int* g)
{
  int i, j, k, r, c, a = (2*n + 1)*(n + 1);
  int *fg = new int[a];
  for (i = 0; i < a; i++){fg[i] = 0;}
  for (i = 0; i < a; i++){
    for (j = 0; j < a; j++){
      r = i/(2*n+1) + j/(2*n+1);
      c = i%(2*n+1) + j%(2*n+1) - 2*n;

```

```
      k = r*(2*n+1) + c+n;
      if (0<=r && r<=n && -n<=c && c<=n){
        fg[k] += f[i] * g[j];
      }
    }
  }
  return fg;
};

int* CalcCmnOfN()
{
  int a = (2*n + 1)*(n + 1), i, c, r, l, k, sign;
  int* genFunc = new int[a];
  int* h = new int[a];
  for (i = 0; i < a; i++){genFunc[i] = 0;}
  genFunc[n] = 1;
  for (r = 1; r <= n; r++){
    for (c = 1; c <= m; c++){
      for (i = 0; i < a; i++){h[i] = 0;}
      l = 0;
      while (l*r <= n){
        sign = 1;
        for (i=n-1; i<=n+1; i++){
          h[i + l*r*(2*n+1)] += sign;
          sign *= -1;
        }
        l += 1;
      }
      genFunc = multiply2(h, genFunc);
    }
  }
  return genFunc;
};

int* crankRes2(int* f)
{
  int i, j, b = p*((n+1)/p), r;
  int *resVector = new int[b];
  for (i = 0; i < b; i++){resVector[i] = 0;}
  for (r = 0; r<(n+1)/p; r++){
    for (i = 0; i < 2*n+1; i++){
      j = i;
      while (j<n){j+=p;}
      resVector[((j-n)%p)+r*p] += f[(n-b+(r+1)*p)*(2*n+1) + i];
    }
  }
  return resVector;
};

int main()
{
  int i, pause;
  int a = (2*n+1)*(n+1);
  int *PowerSeries1 = new int[a];
  int *resVector = new int[n];
  PowerSeries1 = CalcCmnOfN();
  printQZSeries(PowerSeries1);
  resVector = crankRes2(PowerSeries1);
  cout << "\n";
  for (i = 0; i<p*((n+1)/p); i++){
    cout << resVector[i] << "\n";
    if (i%p == p-1){cout << "\n";}
  }

  cin >> pause;
}

```

6.2. Program fmmp. The program fmmp, also written in C++, searches for evidence of linear

congruences for the function $fmmp(n)$. It has a global variable n . First, it calculates $fmmp(a)$ for all a between 0 and n . Then it checks congruences of the form $fmmp(xb + c) \equiv d \pmod{e}$, where all parameters except x are fixed, and x is varied. Only congruences for which there are five x -values with $xb + c \leq n$ are checked. The program outputs all congruences that it cannot falsify. When the program was run, no convincing evidence was found for any linear congruences when $n \leq 50$. Running the program for larger n caused some int type variables to overflow and “loop around.”

```
#include <iostream>
using namespace std;

int n=10;

int* multiply(int* f, int* g)
{
    int i, j;
    int* fg = new int[n];
    for(i=0; i<n; i++){fg[i]=0;}
    for(i=0; i<n; i++){
        for(j=0; j<n; j++){
            if(i+j<n){fg[i+j]+=f[i]*g[j];}
        }
    }
    return fg;
};

int* fmmpOfN()
{
    int i, j, k;
    int* f = new int[n];
    int* g = new int[n];
    f[0]=1;
    for(i=1; i<n; i++){f[i]=0;}
    for(i=1; i<n; i++){
        for(j=1; j<n; j++){
            for(k=1; k<n; k++){g[k]=0;}
            k=0;
            while(i*j*k<n){
                g[i*j*k]++;
                k++;}
            f = multiply(f, g);
        }
    }
    return f;
};

void congruences(int* f)
{
    int a, b, c, m, i, flag;
    for(a=0; a<n/5; a++){
        for(b=0; b<a; b++){
            for(m=2; m<25; m++){
                flag=1;
```

```
for(i=1; i<(n-b)/a; i++){
    if(f[a*i+b]%m!=f[b]%m){flag=0;}
}
if(flag==1){cout <<"fmmp("<<a<<"n"<<b<<"="<<
    f[b]%m<<" (mod"<<m<<"\n");}
}}
};

int main()
{
    int i;
    int pause;
    int* f=new int[n];
    f=fmmpOfN();
    for(i=0; i<n; i++){cout << f[i] <<" ";}
    cin >> pause;
}
```

6.3. Program largeInt. The program largeInt, written in C++, implements a largeInt data type. It has a global variable “size,” and allows the user to declare variables of the largeInt type, which are signed integers with $3 \cdot \text{size}$ digits. Several binary and relational operators are overloaded so that the user may use largeInt variables with the same syntax as int variables in most situations. This code could be included in another program, such as fmmp, to allow for computation with larger numbers. Currently, largeInt does not compute rapidly, and for many computations, one would need to alter it for greater speed. Notably, its division algorithm is especially slow. Presently, the program is set to demonstrate its ability to process large integers by computing 50 factorial.

```
#include <iostream>
using namespace std;

int size=50;

class largeInt
{
public:
    largeInt()
    {
        value=new int[size];
        sign=0;
    };
    largeInt& operator=(int);
    largeInt operator+(largeInt&);
    largeInt operator-(largeInt&);
    largeInt operator*(largeInt&);
    largeInt operator/(largeInt&);
    largeInt operator%(largeInt&);
```

```

largeInt operator+(int);
largeInt operator-(int);
largeInt operator*(int);
largeInt operator/(int);
largeInt operator%(int);
bool operator==(largeInt&);
bool operator<(largeInt&);
bool operator>(largeInt&);
friend ostream& operator<<(ostream&, largeInt&);

int* value;
bool sign;
};
largeInt& largeInt::operator=(int z)
{
int i,d=1;
for(i=0;i<size;i++)
{
if(i<4)
{
this->value[i]=(z/d)%1000;
d*=1000;
}
else
this->value[i]=0;
}
return *this;
}
largeInt largeInt::operator+(largeInt& y)
{
int i, carry;
largeInt z;
carry=0;
for(i=0;i<size;i++)
{
if(value[i]+y.value[i]+carry<1000)
{
z.value[i]=value[i]+y.value[i]+carry;
carry=0;
}
else
{
z.value[i]=value[i]+y.value[i]+carry-1000;
carry=1;
}
}
return z;
}
largeInt largeInt::operator-(largeInt& y)
{
int i, borrow;
largeInt z;
borrow=0;
for(i=0;i<size;i++)
{
if(value[i]-y.value[i]-borrow>=0)
{
z.value[i]=value[i]-y.value[i]-borrow;
borrow=0;
}
else
{
z.value[i]=value[i]-y.value[i]-borrow+1000;
borrow=1;
}
}
return z;
}
largeInt largeInt::operator*(largeInt& y)
{
int i,j,k;
largeInt a,z;
for(i=0;i<size;i++){z.value[i]=0;}
for(i=0;i<size;i++)
{
for(j=0;j<size;j++)
{
for(k=0;k<size;k++){a.value[k]=0;}
if(i+j<size)
a.value[i+j]+=(value[i]*y.value[j])%1000;
if(i+j+1<size)
a.value[i+j+1]+=((value[i]*y.value[j])%1000000)/1000;
if(i+j+2<size)
a.value[i+j+2]+=(value[i]*y.value[j])/1000000;
z=z+a;
}
}
return z;
}
largeInt largeInt::operator/(largeInt& y)
{
largeInt q,q1,m;
q=0;
m=*this;
while(m>y || m==y)
{
q1=1;
while((y*q1)*2<m){q1=q1*2;}
q=q+q1;
m=m-q1*y;
}
return q;
}
largeInt largeInt::operator%(largeInt& y)
{
largeInt q1,m;
m=*this;
while(m>y || m==y)
{
q1=1;
while((y*q1)*2<m){q1=q1*2;}
m=m-q1*y;
}
return m;
}
largeInt largeInt::operator+(int y)
{
largeInt x;
x=y;
return (*this)+x;
}

```

```

largeInt largeInt::operator-(int y)
{
    largeInt x;
    x=y;
    return (*this)-x;
}
largeInt largeInt::operator*(int y)
{
    largeInt x;
    x=y;
    return (*this)*x;
}
largeInt largeInt::operator/(int y)
{
    largeInt x;
    x=y;
    return (*this)/x;
}
largeInt largeInt::operator%(int y)
{
    largeInt x;
    x=y;
    return (*this)%x;
}
bool largeInt::operator<(largeInt& x)
{
    int i=size-1;
    while(value[i]==x.value[i])
    {
        if(i==0)
            return false;
        else
            i--;
    }
    if(value[i]<x.value[i])
        return true;
    else
        return false;
}
bool largeInt::operator>(largeInt& x)
{
    int i=size-1;
    while(value[i]==x.value[i])
    {
        if(i==0)
            return false;
        else
            i--;
    }
    if(value[i]>x.value[i])
        return true;
    else
        return false;
}
bool largeInt::operator==(largeInt& x)
{
    int i;
    bool flag=true;
    for(i=0;i<size;i++)
    {
        if(value[i]!=x.value[i])
            flag=false;
    }
    return flag;
}
ostream& operator<<(ostream& os1, largeInt& x)
{
    int i,j,flag;
    flag=0;
    if(x.sign==1)
        os1<<"-";
    for(i=size-1;i>=0;i--)
    {
        j=100;
        while(j>x.value[i] && j>=1)
        {
            if(flag!=0)
                os1 << 0;
            j/=10;
        }
        if(flag!=0 && x.value[i]!=0)
            os1 << x.value[i];
        if(flag==0 && x.value[i]!=0)
        {
            os1 << x.value[i];
            flag=1;
        }
        if(flag==0)
            os1<<0;
    }
    return os1;
}

int main()
{
    largeInt x;
    x=1;
    int a;
    for(a=2;a<=50;a++)
    {
        x=x*a;
        cout<<x<<"\n";
    }

    int pause;
    cin >> pause;
}

```

7. APPENDIX 2: POETRY

The following poem was composed by the second author for the purpose of harassing his advisor at an REU tea, and thus was never intended to be printed. Therefore the second author apologizes for the following defficiencies: rhyming bat with rat, the entire fifth verse, and the end of the twentieth verse. He does not, however,

apologize for anything else, especially the meter of the poem, as he has never liked meters in the first place, and has never understood how to count syllables properly; therefore any reader that will complain about syllables shall be advised to chew on a brick.

The Mathematician's Bane

(Person A)

In the days of old,
in the mountains of Spain,
there lived a monster;
the mathematician's bane;

T'was eleven feet tall,
with the wings of a bat,
the tail of a scorpion,
the teeth of a rat;

On the head of a man,
were the horns of a goat,
and the body of a lion
had a bright purple coat.

And every fifth year,
on the solstice of June,
he came down from the hills,
to bring the village's ruin.

Though terrible in body,
more sinister was his mind,
and he challenged the people,
to bring what hero they'd find;

To answer a riddle,
to measure his wit,
and the fate of the town
would on razor's edge sit.

And for our hero,
to fail was to die,
and in poisonous smoke,
the town would fry;

But should the riddle be answered,
the demon must be gone,
and leave in peace,
for five years on.

And every time,
T'was the hero's fate sealed,
because no answer,
could be revealed:

Name for me
the largest prime;
Or square a circle
by compass and line;

And tell me now
the last digit of pi;
no riddle had an answer,
so each hero did die.

But forever now
in song expound--
that by sweet chance,
a hero was found;

Brazen in courage,
unmatched in wit;
came before the monster,
and impertently spit

taunts and challenges,
enraged the beast;
of the hero's thoughts,
fear was least.

And the demon was furious,
like never before,
and into this riddle,
all his malice did pour;

If you answer this riddle,
I will be destroyed;
my body will burn,

and my spirit made void.

But if you should fail,
then the world will rue,
the terrible things,
I shall certainly do:

I'll flood each town
in the country of Spain,
and burn to the ground
what ruins remain;

The mountains will spawn
rivers of fire,
and pillars of smoke
to the stars and higher;

The land will be barren
for seventy-one years,
and the country overrun
with terrible fears;

and just when you think
my mischief is done,
to top it all off,
I'll burn out the sun.

If you challenge me thus,
you must answer me then:
how many are
the partitions of n?

(Person B)

$$p(n) = \frac{1}{\pi\sqrt{2}} \sum_{k=1}^{\infty} A_k(n) \sqrt{k} \frac{d}{dn} \left(\frac{\sinh\left(\frac{\pi}{k} \sqrt{\frac{2}{3} \left(n - \frac{1}{24}\right)}\right)}{\sqrt{n - \frac{1}{24}}} \right)$$

where

$$A_k(n) = \sum_{\substack{0 \leq m < k \\ \gcd(m,k)=1}} \exp(\pi i \cdot s(m,k) - 2\pi i n m / k)$$

and s(m,n) indicates a Dedekind sum.

(Person A)
In a flash of smoke,
and a blast of sound,
the monster was gone;
no trace to be found.

And the hero was saved,
and so was the town,
and for the hero,
they made a crown;

And inscribed on the crown,
made for the hero,
was the identity
e to the pi i plus one equals zero.

REFERENCES

- [1] Andrews, George E. *Number Theory*. Dover Publications: New York, 1971.
- [2] Andrews, George E. *A Survey of Multipartitions: Congruences and Identities*, preprint.
- [3] Andrews, George E. *The Theory of Partitions*. Encyclopedia of Mathematics and Its Applications. Addison-Wesley Publishing Co.: Reading, MA, 1976.
- [4] Andrews, George E. *Dyson's Crank of a Partition*. Bulletin (New Series) of the American Mathematical Society. Vol. 18, No. 2, April 1988.

WISCONSIN LUTHERAN COLLEGE

E-mail address: joanna.furno@wlc.edu

OREGON STATE UNIVERSITY

E-mail address: waterspa@onid.orst.edu