THE DAVENPORT CONSTANT AND AUTOMORPHICALLY EQUIVALENT ELEMENTS

ARJUN AGARWAL, RACHEL CHEN, AND ROHAN GARG

Abstract. Let G be a finite abelian group and let $D(G)$ be the Davenport constant of the group. In this paper we demonstrate several bounds on the Davenport constant. We also investigate whether $D(G)$, along with other numerical invariants of the group, is sufficient to uniquely determine its structure. Our investigations lead us to a conjecture that relates the divisibility of the Davenport constant of the subgroups to the structure of the group. We also study the inverse Davenport problem– the structure of maximal 0-sequences of length $D(G)$. The structure of these sequences motivates the study of necessary and sufficient conditions for two elements x and $y \in G$ to be automorphic images of one another. We ultimately prove that there exists $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$ if and only if $G/\langle x\rangle \cong G/\langle y\rangle$. This result leads to our development of the two fastest known algorithms to determine if two elements of a finite abelian group are automorphic images of one another. We use this algorithm to develop the fastest known algorithm to compute the orbits of finite abelian groups.

1. Introduction

In this paper, we will study the Davenport constant and areas related to it, such as the inverse Davenport problem and conditions on when two elements of a finite abelian group are automorphic images of one another.

First introduced by Rogers (see [15]) in 1963, and made famous by Harold Davenport at the 1966 Conference in Group Theory and Number Theory [13], the Davenport constant has become a classic object of study at the intersection of algebra, combinatorics, and number theory. Defined as the minimal number n such that any G-sequence of length n must have a subsequence summing to the identity, this constant is important because of its use in the study of factorization in algebraic number rings. Given a Dedekind domain D, its *elastisicity* (denoted $\rho(D)$), a measure of failure of unique factorization lengths, is closely related to the Davenport constant of the ideal class group $G \cong \mathrm{Cl}(D)$ of D: $\rho(D) \leqslant \frac{D(G)}{2}$ with equality achieved if G is finite and each ideal class in G contains a prime of D (see [12, 20]). In the proof of the existence of infinitely many Carmichael numbers, this bound plays an important role [1].

In Section 3, we start the paper by stating additional bounds on $D(G)$ that do not appear to be in the literature. In Section 4, we investigate the relation of $D(G)$ with the structure of the finite abelian group G. We propose an original problem on the relation of the Davenport constant of the subgroups of G to $D(G)$ and solve it for numerous cases.

We then shift gears to the inverse Davenport problem, the investigation of the structure of maximal 0-sequences of length $D(G)$. We propose and investigate definitions for equivalence classes of sequences, contrary to previous papers which were more focused on solving the inverse Davenport problem completely for specific classes of groups (see

[4, 9, 16]). These equivalence classes motivate us to investigate when two elements x, $y \in G$ are automorphic images of each other.

Finally, in Section 7, we state and prove a theorem that is important not just in the study of the Davenport constant, but in group theory in general: given $x, y \in G$, there exists $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$ if and only if $G/\langle x\rangle \cong G/\langle y\rangle$. In Section 8, we use this result to develop of the two fastest known algorithms to determine if two elements of a finite abelian group are automorphic images of one another by computing matrices in Smith Normal Form. We use these algorithms to compute the orbits of finite abelian groups in Section 9, in the fastest known time complexity.

2. Background and Literature Review

Let G be a finite abelian group, written additively.

Definition 2.1. A G-sequence $\{g_1, g_2, \ldots, g_n\}$ of (not necessarily distinct) elements of G is a *0-sequence* (alternatively *zero-sum sequence*) if $g_1 + g_2 + \cdots + g_n = 0$. We say that the sequence $\{g_1, g_2, \ldots, g_n\}$ has a *0-subsequence* (alternatively *zero-sum subsequence*) if there exists a nonempty set $S \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in S} g_i = 0$.

Definition 2.2. The *Davenport constant*, $D(G)$, of G is defined as

$$D(G) = \min\{n \mid \text{every } G\text{-sequence of length } n \text{ has a 0-subsequence}\}.$$

Alternatively, the Davenport constant can also be defined as follows.

Definition 2.3. $D(G)$ is the maximum length of a 0-sequence that contains no proper zero subsequence.

Proposition 2.4. Definition 2.2 and Definition 2.3 are equivalent.

Proof. Let x be the value of $D(G)$ as in Definition 2.2, let y be $D(G)$ as in Definition 2.3.

First, we prove that $x \geqslant y$. Let A be a 0-sequence with length y and no proper 0-subsequences. If $x < y$, any x-element subsequence in A must have a 0-subsequence, so A contains a proper 0-subsequence, which is a contradiction.

We finish by proving that $y \geqslant x$. There exists a G-sequence $\{g_1, g_2, \ldots, g_{x-1}\}$ with no 0-subsequence by definition. Let $g_x = -(g_1 + g_2 + \cdots + g_{x-1})$. The sequence $A = \{g_1, g_2, \ldots, g_{x-1}, g_x\}$ is a 0-sequence. Assume A contains a proper 0-subsequence. Note that this 0-subsequence must contain g_x since the original sequence had no zero-subsequences. However, taking every term in A that does not appear in this 0-subsequence creates a 0-subsequence in $\{g_1, g_2, \ldots, g_{x-1}\}$, which is a contradiction, so A is a 0-sequence with no proper 0-subsequences. Therefore, $|A| = x \leqslant y$.

We conclude that $x = y$, so the two definitions are equivalent. □

Definition 2.5. We say a 0-sequence is *maximal* if it contains no proper 0-subsequences.

Definition 2.6. Throughout this paper, given $x \in G$ and $k \in \mathbb{N}$, let kx denote $\underbrace{x + x + \cdots + x}_{k \text{ times}}$.

If $C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_t}$ is the invariant factor decomposition of G, consider the sequence consisting of $(1, 1, 1, \ldots 1)$ and for each $1 \leqslant i \leqslant t$, we take $m_i - 1$ copies of the element that has a 0 in every component except the i^{th} component which is 1. This sequence is a maximal 0-sequence and has length $m_1 + \ldots m_t - t + 1$, which motivates the following definition.

Definition 2.7. Let $G \cong C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_t}$ be a finite abelian group expressed in its invariant factor decomposition. Define $D^*(G) = 1 + \sum_{i=1}^{t}(m_i - 1)$.

From Definition 2.3, it follows that $D(G) \geqslant D^*(G)$; the question of when equality is achieved is a central question in the study of the Davenport constant.

From [13], we have $D(G) \leqslant |G|$. The following is a much tighter bound of the Davenport constant.

Theorem 2.8 (Emde-Boas [21])**.** Given a finite abelian group $G \cong C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_t}$ written in its invariant factor decomposition,

$$D(G) \leqslant \exp(G)\left(1 + \log\left(\frac{|G|}{\exp(G)}\right)\right) = m_t\left(1 + \log\left(\frac{m_1 m_2 \dots m_t}{m_t}\right)\right).$$

This bound is used frequently throughout this paper.

While a formula for the Davenport constant is not known for a general abelian group, it is known for some special cases.

The following is the list of groups where it is known that $D(G) = D^*(G)$ from [10, 17]:

- G has rank less than or equal to 2;
- G is a p-group;
- $G \cong G_1 \oplus C_{p^k n}$, where G_1 is a p-group and $p^k \geqslant D^*(G_1)$;
- $G \cong C_2^3 \oplus C_{2n}$ where n is odd;
- $G \cong C_{2p^{k_1}} \oplus C_{2p^{k_2}} \oplus C_{2p^{k_3}}$ where p is prime;
- $G \cong C_2 \oplus C_{2n}^2$ with $p \nmid n$ for every prime $p \geqslant 11$;
- $G \cong C_3 \oplus C_3 \oplus C_{3d}$ where $d \in \mathbb{N}$;
- $G \cong C_{3\cdot 2^t} \oplus C_{3\cdot 2^u} \oplus C_{3\cdot 2^v}$ where $t \geq u \geq v$;
- $G \cong C_4 \oplus C_4 \oplus C_{4d}$ where $d \in \mathbb{N}$;
- $G \cong C_6 \oplus C_6 \oplus C_{6d}$ where $d \in \mathbb{N}$.

However, it is known that in general, $D(G) \neq D^*(G)$. One such class of groups is shown below.

Theorem 2.9. (Geroldinger and Schneider [10]) Let $n \geqslant 2, k \geqslant 2$ with $\gcd(n, k) = 1$, $0 \leqslant \rho \leqslant n - 1$, and $G = C_n^{(k-1)n+\rho} \oplus C_{kn}$.

(1) If $\rho \geqslant 1$ and $\rho \not\equiv n \pmod{k}$, then $D(G) \geqslant D^*(G) + \rho$.
(2) If $\rho \leqslant n - 2$ and $x(n - \rho + 1) \not\equiv n \pmod{k}$ for any $x \in \{1, 2, \dots, n - 1\}$, then $D(G) \geqslant D^*(G) + \rho + 1$.

3. Additional Bounds on $D(G)$

The information in the previous section is all known in existing literature. In this section, we will develop additional bounds.

Proposition 3.1. Let $G_1, G_2, \dots, G_n$ be finite abelian groups. Then, $D(G_1 \oplus G_2 \oplus \cdots \oplus G_n) \leqslant \prod_{i=1}^{n} D(G_i)$.

Proof. We will first prove that if G, H are finite abelian groups then $D(G \oplus H) \leqslant D(G)D(H)$. Let $D(G) = k$ and $D(H) = m$; let S be a sequence of km elements from $G \oplus H$, so $S = \{(g_1, h_1), (g_2, h_2), \dots, (g_{km}, h_{km})\}$ where $g_i \in G$ and $h_i \in H$.

We define m sequences S_1, S_2, …, S_m, where the sequence S_i consists of the i^{th} "block" of k elements, or elements $i(k-1)+1$ to ik. Consider S_i for all i. Since $D(G) = k$, S_i

contains a subsequence K_i where the corresponding elements from G (the first component of the elements) sum to zero. Let the sum of the corresponding elements of S_i from H (the sum of the elements in the second component) be p_i; there are m such sums. Since $D(H) = m$, the sequence $\{\{p_1, p_2, \ldots, p_m\}$ where $p_i \in H\}$ contains a 0-subsequence. By construction, the corresponding elements from G for each of these sequences already form a 0-subsequence. Therefore, there is a 0-subsequence in S. This proves that $D(G \oplus H) \leqslant D(G)D(H)$.

Inductively applying this property to $G_1 \oplus G_2 \oplus \cdots \oplus G_n$, we get $D(G_1 \oplus G_2 \oplus \cdots \oplus G_n) \leqslant \prod_{i=1}^{n} D(G_i)$. □

Since the above bound is very weak for groups of higher rank, it is worth exploring if a tighter bound can be established.

Proposition 3.2. If G is a finite abelian group of rank $d \geqslant 2$, we have $D(G) \leqslant |G| - d + 1$.

Proof. We will induct on the rank of G. Let $G \cong C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_d}$ with $m_1 \mid m_2 \mid \cdots \mid m_d$. The base case, rank 2, is true since $D(G) = m_1 + m_2 - 1$ and $m_1 + m_2 - 1 \leqslant m_1 m_2 - 1$ for all $m_1, m_2 \geqslant 2$, since we can rewrite this as $(m_1 - 1)(m_2 - 1) \geqslant 1$.

By Proposition 3.1 and the inductive hypothesis,

$$\begin{aligned} D(G) \leqslant D(C_{m_d}) \cdot D(C_{m_1} \oplus \cdots \oplus C_{m_{d-1}}) &\leqslant m_d(m_1 m_2 \ldots m_{d-1} - (d-1) + 1) \\ &= |G| - m_d(d-2). \end{aligned}$$

Note that $m_d(d-2) \geqslant d-1$ for $d > 2$, $m_d \geqslant 2$. □

Remark 3.3. For rank $d > 3$, $|G| - m_d(d-2)$, where m_d is the exponent of the group, is an even better bound than the one above.

We will now investigate some bounds for the quotient group G/H where G is a finite abelian group and H is a subgroup of G.

Theorem 3.4. Given finite abelian group G and subgroup $H \leq G$, $D(G/H) \leq D(G)$.

Proof. Consider the sequence $\overline{g_1}, \overline{g_2}, \ldots, \overline{g_k}$ of elements from G/H and consider the sequence $g_1, g_2, \ldots, g_k$ in G. If $k \geqslant D(G)$ then there exists a 0-subsequence $g_{i_1}, g_{i_2}, \ldots, g_{i_m}$. The sequence $\overline{g_{i_1}}, \overline{g_{i_2}}, \ldots, \overline{g_{i_m}}$ in G/H will be a 0-sequence because

$$\overline{g_{i_1}} + \overline{g_{i_2}} + \cdots + \overline{g_{i_m}} = \overline{g_{i_1} + g_{i_2} + \cdots + g_{i_m}} = \bar{0},$$

implying $D(G/H) \leqslant D(G)$. □

4. The Relation Between $D(G)$ and the Structure of G

From the previous sections, we know that if G is cyclic, $D(G) = |G|$. It is natural to ask if the converse also holds true, which we prove in the following proposition.

Proposition 4.1. $D(G) = |G|$ if and only if G is cyclic.

Proof. If G is cyclic and g is a generator of G, the sequence $\{g, g, \ldots, g\}$ where g is repeated $|G|$ times is a maximal 0-sequence. We have already established that $D(G) \leqslant |G|$, so we are done.

For the other direction, assume $D(G) = |G|$ and we prove G is cyclic. Let $|G| = n$ and consider a maximal 0-sequence $A = \{a_i\}_{i=1}^{n}$ and define $b_i = a_1 + a_2 + \cdots + a_i$ for $1 \leqslant i \leqslant n$. Because A is a maximal 0-sequence, all terms of $\{b_i\}_{i=1}^{n}$ must be distinct. We swap the

order of a_1 and a_2 to create a maximal 0-sequence $A' = \{a_2, a_1, \ldots, a_n\} = \{a'_1, a'_2, \ldots, a'_n\}$. Define $b'_k = \sum_{i=1}^{k} a'_i$ for $1 \leqslant k \leqslant n$. Note that $b_k = a_1 + a_2 + \cdots + a_k = a_2 + a_1 + \cdots + a_k = a'_1 + a'_2 + \cdots + a'_k = b'_k$ for all $k > 1$, so $b_1 = b'_1$, implying $a_1 = a_2$. Without loss of generality this proves that $a_i = a_j$ for $1 \leqslant i, j \leqslant n$ since the choice of a_1 and a_2 was arbitrary, implying that G is generated by a_1 and that the group is cyclic. □

Clearly, $D(G)$ is determined by G. It is natural to ask if $D(G)$, along with other numerical invariants of G, can uniquely identify G.

Question 4.2. Let G be a finite abelian group. Is knowing $|G|$ and $D(G)$ sufficient to determine G?

We give a counterexample with two groups, G_1 and G_2, where $|G_1| = |G_2|$ and $D(G_1) = D(G_2)$. We can find a counterexample in 2-groups. Consider $G_1 = C_2 \oplus C_{2^2}^4 \oplus C_{2^3}$. Then $|G_1| = 2^{1+2+2+2+2+3} = 2^{12}$. Since G_1 is a 2-group, we have $D(G_1) = (2-1) + 4 \cdot (2^2 - 1) + (2^3 - 1) + 1 = 21$. Let $G_2 = C_2^6 \oplus C_{2^3}^2$. Then $|G_2| = 2^{1+1+1+1+1+1+3+3} = 2^{12}$. Since G_2 is a 2-group, we have $D(G_2) = 6 \cdot (2-1) + 2 \cdot (2^3 - 1) + 1 = 21 = D(G_1)$. The following question adds a new condition for this question, motivated by the above counterexample.

Question 4.3. If the rank, the order, and Davenport constant of a finite abelian group G is known, can G be uniquely determined?

The answer to the above question is no. There are infinitely many pairs of non-isomorphic groups which have the same rank, order, and Davenport constant.

Proposition 4.4. Consider the p-groups G_1 and G_2 defined by

$$G_1 = C_p^p \oplus C_{p^3}^{p+2} \text{ and } G_2 = C_{p^2}^{2p+1} \oplus C_{p^4}.$$

We have $|G_1| = |G_2|$, $D(G_1) = D(G_2)$, $\operatorname{rank}(G_1) = \operatorname{rank}(G_2)$, but $G_1 \ncong G_2$.

Proof. We have $\operatorname{rank}(G_1) = \operatorname{rank}(G_2) = 2p + 2$ and $|G_1| = p^p \cdot p^{3(p+2)} = p^{4p+6} = p^{2(2p+1)}p^4 = |G_2|$. Since G_1 and G_2 are p-groups, $D(G_1) = p(p-1) + (p+2)(p^3-1) + 1 = p^4 + 2p^3 + p^2 - 2p - 1$. Also, $D(G_2) = (2p+1)(p^2-1) + (p^4-1) + 1 = p^4 + 2p^3 + p^2 - 2p - 1$. Therefore, $D(G_1) = D(G_2)$ but $G_1 \ncong G_2$. □

The above construction is not unique.

Some additional observations can be made about pairs of non-isomorphic groups that have the same rank, order, and Davenport constant.

Theorem 4.5. For every prime p, there exists an infinite number of non-isomorphic p-groups G_1 and G_2 satisfying $D(G_1) = D(G_2)$, $|G_1| = |G_2|$, $\operatorname{rank}(G_1) = \operatorname{rank}(G_2)$, but $G_1 \ncong G_2$.

Proof. Consider the two p-groups G_1 and G_2 constructed in Proposition 4.4. We can replace each invariant factor C_{p^k} by $C_{p^{k+1}}$ to get a new pair of groups that satisfies our conditions. Consider

$$G_1 = C_{p^{1+n}}^p \oplus C_{p^{3+n}}^{p+2} \text{ and } G_2 = C_{p^{2+n}}^{2p+1} \oplus C_{p^{4+n}} \quad \text{where } n \in \mathbb{N}.$$

Then, for all $n \in \mathbb{N}$, we have $D(G_1) = D(G_2)$, $|G_1| = |G_2|$, $\operatorname{rank}(G_1) = \operatorname{rank}(G_2)$, but $G_1 \ncong G_2$, showing that there is an infinite number of such p-groups. Another way of extending the groups is by considering $G'_1 = C_{p^k} \oplus G_1$ and $G'_2 = C_{p^k} \oplus G_2$. If G_1 and G_2 satisfy the properties above then so do G'_1 and G'_2. In fact, we can generalize the

construction above even more. Define $G_{1n} = C_{p^{1+n}}^{p} \oplus C_{p^{3+n}}^{p+2}$ and $G_{2n} = C_{p^{2+n}}^{2p+1} \oplus C_{p^{4+n}}$. Then the groups $G_1 = \prod_{i=1}^{t} G_{1n_i}$ and $G_2 = \prod_{i=1}^{t} G_{2n_i}$ also satisfy $D(G_1) = D(G_2)$, $|G_1| = |G_2|$, $\text{rank}(G_1) = \text{rank}(G_2)$, but $G_1 \ncong G_2$. □

Question 4.6. Suppose there is a finite abelian group G with $D(G) > D^*(G)$. Is this also true for all finite abelian groups that contain G?

The answer is no. Consider $G = C_2^4 \oplus C_6$, which has $D(G) > D^*(G)$ due to Theorem 2.9, as we can set $n = 2$, $k = 3$, and $\rho = 0$. However, $C_2^4 \oplus C_{2d}$ where $d \geqslant 70$ is even has $D(G) = D^*(G)$, according to Theorem 5.8 in [4].

This is a very interesting result as it implies that there are groups for which $D(G) = D^*(G)$ but $D(H) > D^*(H)$ for some subgroup H. If we think about modifying H (by either adding more invariant factors or increasing existing ones) with $D(H) > D^*(H)$ to get a group G, when can this"repairing" or "healing" effect take place? The next theorem addresses this question.

Theorem 4.7. Let G be a finite abelian group with $D(G) > D^*(G)$ and $G \leq H$. Let G be a direct summand of H such that the invariant factors of G appear as invariant factors of H also. Then, $D(H) > D^*(H)$.

Proof. Let $\text{rank}(H) = t$ and write the invariant factor decomposition $H \cong C_{a_1} \oplus C_{a_2} \oplus \cdots \oplus C_{a_t}$ where $1 < a_1 \mid a_2 \mid \cdots \mid a_t$. Since G is a direct summand of H, we have $H \cong G \oplus K$ where K is a non-trivial abelian group. Therefore, there exists $A \subset \{1, 2, \ldots, t\} = [t]$ such that $G \cong \bigoplus_{i \in A} C_{a_i}$ and $K \cong \bigoplus_{j \in [t]-A} C_{a_j}$. Because $D(G) > D^*(G)$ there exists a G-sequence $\{g_1, \ldots, g_n\}$ with $n \geqslant D^*(G)$, which contains no 0-sequence. Now let our current H-sequence be

$$S_H = \{(g_1, 0), \ldots, (g_n, 0)\},$$

where each component corresponds to an element of G and K and 0 represents the identity in K. Assume for the sake of contradiction that $D(H) = D^*(H)$. Then $D^*(H) = D^*(G) + D^*(K) - 1$ implies that $D(H) = D^*(G) + D^*(K) - 1$. We will now expand the sequence S_H as follows. For each $j \in [t] - A$ consider the element $h_j \in H$ consisting of the t-tuple with 0 everywhere except for the j^{th} entry where we have 1. This element has order a_j and so we include it $(a_j - 1)$ times in S_H. Now, we have a sequence S_H satisfying

$$|S_H| = n + \sum_{j \in [t]-A} (a_j - 1) = n + D^*(K) - 1 \geq D^*(G) + D^*(K) - 1 = D(H).$$

This implies that we have a 0-sequence in S_H, but the way we constructed S_H makes this impossible. Therefore, $D(H) \neq D^*(H)$ and indeed, $D(H) > D^*(H)$, as desired. □

5. The $D(H) \mid D(G)$ Problem

In this section, we expand on the question of how $D(G)$ relates to the structure of G by investigating how $D(G)$ relates to the Davenport constant of the subgroups of G.

One natural question to ask is: for which groups G do all of its subgroups H satisfy $D(G/H) = D(G)/D(H)$?

Lemma 5.1. An abelian group G satisfies $D(G/H) = D(G)/D(H)$ for each $H \leqslant G$ if and only if G is cyclic.

Proof. Let G be represented by its invariant factor decomposition $G \cong C_{m_1} \oplus C_{m_2} \oplus \dots C_{m_t}$. Consider the subgroup $H \cong C_{m_t}$. Then we know $G/H \cong C_{m_1} \oplus C_{m_2} \oplus \dots C_{m_{t-1}}$. Since $D(G) \geq D^*(G)$, we have

$$D(G/H) \geq 1 + \sum_{i=1}^{t-1}(m_i - 1).$$

We claim that if $D(G/H) = D(G)/D(H)$, then G has rank 1. Assume $D(G/H) = D(G)/D(H)$. From Theorem 2.8, $D(G) \leq m_t + m_t \log(m_1 m_2 \dots m_{t-1})$. This implies that

$$\begin{aligned} D(G)/D(H) &\leq \frac{m_t + m_t \log(m_1 m_2 \dots m_{t-1})}{D(H)} \\ &= \frac{m_t + m_t \log(m_1 m_2 \dots m_{t-1})}{m_t} \\ &= 1 + \log(m_1 m_2 \dots m_{t-1}). \end{aligned}$$

Combining all the inequalities, we get

$$1 + \sum_{i=1}^{t-1}(m_i - 1) \leq D(G/H) = D(G)/D(H) \leq 1 + \log(m_1 m_2 \dots m_{t-1}).$$

which rearranges into

$$\sum_{i=1}^{t-1} \log(m_i) \geq \sum_{i=1}^{t-1}(m_i - 1).$$

We know $m_i > 1 + \log(m_i)$ for all $m_i > 1$. Therefore, this inequality does not hold for $t > 1$, and t must be equal to 1.

If $t = 1$, let $G \cong C_{m_1}$. Consider a subgroup $H \cong C_m$ with $m \mid m_1$. Then

$$D(G/H) = D(C_{m_1/m}) = m_1/m = D(G)/D(H),$$

as desired. □

We can loosen the above constraint and investigate the following.

Definition 5.2 (Property P)**.** Define finite abelian group G to have Property P if for all subgroups $H \leqslant G$, $D(H) \mid D(G)$.

It is straightforward to show that cyclic groups have Property P. We conjecture that the only groups that have Property P are cyclic groups. In this section, we eliminate some cases.

Theorem 5.3. If G is rank 2, G does not have Property P.

Proof. Let $G \cong C_a \oplus C_{ab}$. Then $D(G) = a + ab - 1$. However, the subgroup $H = C_a$ satisfies $D(H) = a$ and $a \nmid a + ab - 1$, so G does not satisfy the condition. □

Theorem 5.4. If G is rank 3, G does not have Property P.

Proof. Let $G \cong C_a \oplus C_{ab} \oplus C_{abc}$ and assume G has Property P. By Theorem 2.8, we know that

$$D(G) \leqslant abc\left(1 + \log\left(\frac{a \cdot ab \cdot abc}{abc}\right)\right) = abc(1 + \log(a) + \log(ab)).$$

Consider the subgroups C_{ab} and $C_{ab} \oplus C_{abc}$. The Davenport constants of these two groups are ab and $ab + abc - 1$ respectively. If the hypothesis is true,

$$ab \mid D(G) \quad \text{and} \quad ab + abc - 1 \mid D(G).$$

Thus, $\text{lcm}(ab, ab+abc-1) \mid D(G)$. Since $\gcd(ab, ab+abc-1) = 1$, we have $ab(ab+abc-1) \mid D(G)$ which implies

$$ab(ab + abc - 1) \leqslant abc(1 + \log(a) + \log(ab)).$$

This simplifies to $ab + abc - 1 \leqslant c + c\log(a) + c\log(ab)$. We can rewrite this (where e refers to Euler's number) as

$$ab - 1 \leqslant c(\log(a^2be) - ab).$$

Using a computational tool, such as Wolfram Alpha, we find this inequality never holds if $b > 1$ and that we must have $a = 2$ or 3 ($a = 1$ is not allowed, since C_1 can't be one of the factors) for the inequality to hold.

We solve each case separately. If $a = 2$, we can write the group as $G = C_2 \oplus C_2 \oplus C_{2c}$. According to [6], we have $D^*(G) \leqslant D(G) \leqslant D^*(G) + 1$ for groups of the form $C_n \oplus C_{nm} \oplus C_{nmq}$ for $n = 2$ or 3. We have $D^*(G) = 2 + 2c$, so $2 + 2c \leqslant D(G) \leqslant 3 + 2c$. But $C_2 \oplus C_{2c}$ is a subgroup of G, so $2c + 1 \mid D(G)$. This is impossible, since $\gcd(2c + 1, 2c + 2) = 1$ and $\gcd(2c + 1, 2c + 3) = \gcd(2c + 1, 2) = 1$, so $2c + 1 \nmid 2c + 2$ and $2c + 1 \nmid 2c + 3$ for all $c \geqslant 1$.

We now consider the case of $a = 3$. Let $G = C_3 \oplus C_3 \oplus C_{3c}$. From the list in Section 2, $D(G) = D^*(G)$ in this group. Because $D^*(G) = 3c+4$ and $3 = D(C_3) \mid D(G) = D^*(G) = 3c + 4$, which is impossible, we have reached a contradiction.

Having exhausted all cases, we have proven that if G is rank 3, then the condition does not hold. □

Theorem 5.5. If G is rank 4, G does not have Property P.

Proof. Let $G \cong C_a \oplus C_{ab} \oplus C_{abc} \oplus C_{abcd}$ with $a \geqslant 2$. Since $D(A \oplus B) \leqslant D(A)D(B)$ due to Proposition 3.1, we can write

$$D(G) \leqslant (a + ab - 1)(abc + abcd - 1).$$

Consider the two subgroups $H_1 = C_{abc}$ and $H_2 = C_{abc} \oplus C_{abcd}$. We know that $D(H_1) = abc$ and $D(H_2) = abc + abcd - 1$. Since both of these values divide $D(G)$ and are relatively prime, we get $abc(abc + abcd - 1) \mid D(G)$ which implies that

$$abc(abc + abcd - 1) \leqslant D(G) \leqslant (a + ab - 1)(abc + abcd - 1).$$

We can divide out $abc + abcd - 1$ to get $abc \leqslant a + ab - 1$. If $c > 1$, we get that $ab + ab \leqslant abc \leqslant a + ab - 1$, but this is clearly false as $ab \geqslant a$ and $ab \geqslant ab$, so $2ab \geqslant a + ab$ which implies $2ab > a + ab - 1$.

We are left to consider $c = 1$. We use the Emde-Boas bound (Theorem 2.8) to get that

$$ab(ab + abd - 1) \leqslant D(G) \leqslant abd(1 + \log(a^3b^2))$$

which implies that

$$\frac{ab - 1}{d} + ab \leqslant 1 + \log(a^3b^2).$$

We claim that for $a > 6$, $ab > 1 + \log(a^3b^2) = 1 + 3\log a + 2\log b$.

Notice that increasing b by one increases the left hand side by a while it increases the right hand side by at most $2\log 2 < 2$. Therefore, if the inequality holds true for some (a, b), it must hold true for (a, x) for all $x \geqslant b$.

When $b = 1$, we get $a - 3\log a - 1 > 0$. The derivative of this is $1 - \frac{3}{a}$. For $a > 3$, the derivative is positive, so the function increases. At $a = 7$, we get $7 - 3\log 7 - 1 > 0$, so for all $a > 6$, the inequality must hold.

We are left to consider $a < 7$. We find all ordered pairs (a, b) that don't satisfy $ab > 1 + \log(a^3b^2) = 1 + 3\log a + 2\log b$:

- If $a = 2$, we can check that $(2, 3)$ satisfies the inequality and $(2, 2)$ does not, so the only ordered pairs here are $(2, 1)$ and $(2, 2)$;
- If $a = 3$, we can check that $(3, 2)$ satisfies the inequality and $(3, 1)$ does not, so the only ordered pair here is $(3, 1)$;
- If $a = 4$, we can check that $(4, 2)$ satisfies the inequality and $(4, 1)$ does not, so the only ordered pair here is $(4, 1)$;
- If $a = 5$, we can check that $(5, 2)$ satisfies the inequality and $(5, 1)$ does not, so the only ordered pair here is $(5, 1)$;
- If $a = 6$, we can check that $(6, 2)$ satisfies the inequality and $(6, 1)$ does not, so the only ordered pair here is $(6, 1)$.

We now prove that none of these work using the Emde-Boas bound.

For $(2, 1)$, we are considering groups of the form $G \cong C_2 \oplus C_2 \oplus C_2 \oplus C_{2d}$. We know that $D(G) \leqslant 2d(1 + \log 8) \leqslant 7d$. But the subgroups $C_2 \oplus C_2 \oplus C_2$ and $C_2 \oplus C_{2d}$ have Davenport constants 4 and $2d + 1$ respectively, which are relatively prime. So $8d + 4 = 4(2d + 1) \leqslant D(G) \leqslant 7d$, which is not true.

For $(2, 2)$, the proof is analogous to $(2, 1)$, we use the subgroups $C_2 \oplus C_2 \oplus C_4$ and $C_4 \oplus C_{4d}$ which have Davenport constants 8 and $4d + 3$, which are relatively prime.

For $(3, 1)$, the proof is analogous to $(2, 1)$, we use the subgroups $C_3 \oplus C_3 \oplus C_3 \oplus C_3$ and $C_3 \oplus C_{3d}$ which have Davenport constants 9 and $3d + 2$, which are relatively prime.

For $(4, 1)$, the proof is analogous to $(2, 1)$, we use the subgroups $C_2 \oplus C_4 \oplus C_4$ and $C_4 \oplus C_4$ which have Davenport constants 8 and $4d + 3$, which are relatively prime.

For $(5, 1)$, we are considering groups of the form $G \cong C_5 \oplus C_5 \oplus C_5 \oplus C_{5d}$. First note that $5(5d + 4) \leqslant 5d(1 + \log 125)$ by the Emde-Boas bound. Solving the inequality, we get $d \geqslant 5$. Now consider the two subgroups C_{5d} and $C_5 \oplus C_{5d}$. The Davenport constants are $5d$ and $5d + 4$ respectively. Note that $\gcd(5d, 5d + 4) \mid 4$, so $\operatorname{lcm}(5d, 5d + 4) \geqslant \frac{5d(5d+4)}{4}$. Then $\frac{5d(5d+4)}{4} \leqslant 5d(1 + \log 125)$. Solving the quadratic shows that this inequality does not hold for $d \geqslant 5$, as desired.

For $(6, 1)$, the proof is analogous to $(2, 1)$, we use the subgroups $C_3 \oplus C_3 \oplus C_3 \oplus C_3$ and $C_6 \oplus C_{6d}$ which have Davenport constants 9 and $6d + 5$, which are relatively prime. □

Lemma 5.6. If G is a p-group of rank greater than 1, G does not have Property P.

Proof. Let the invariant factor decomposition of G be $G \cong C_{p^{a_1}} \oplus C_{p^{a_2}} \oplus \cdots \oplus C_{p^{a_k}}$ with $k > 3$ (we already resolved all groups with rank 2 and 3) and assume it has Property P. The Davenport constant of G is $-k + 1 + \sum_{i=1}^{k} p^{a_i}$ by [13]. Consider the subgroup $H = C_{p^{a_2}} \oplus \cdots \oplus C_{p^{a_k}}$. It has Davenport constant $-k + 2 + \sum_{i=2}^{k} p^{a_i}$. Note this is strictly

less than $D(G)$ and divides $D(G)$; therefore, it is at most $D(G)/2$. We have,

$$2D(H) = -2k + 4 + 2\sum_{i=2}^{k} p^{a_i}.$$

However, note that

$$2\sum_{i=2}^{k} p^{a_i} - \sum_{i=1}^{k} p^{a_i} \geqslant p^{a_1} \cdot (k-2) \geqslant 2(k-2) > k-3 = 2(k-2) + (1-k),$$

which can be rewritten as

$$2D(H) = 2(-k+2) + 2\sum_{i=2}^{k} p^{a_i} > (-k+1) + \sum_{i=1}^{k} p^{a_i} = D(G)$$

which gives a contradiction. □

Lemma 5.7. If G is a group of at least rank 2 with all distinct invariant factors, then there exists at least one subgroup $H \leqslant G$, such that $D(H) \nmid D(G)$.

Proof. We need to show that G does not have Property P. Let $G \cong C_{a_1} \oplus C_{a_1a_2} \oplus \cdots \oplus C_{a_1a_2\ldots a_n}$ satisfy Property P. Define $P_{i,j} = a_i a_{i+1} \cdots a_j$.

By Theorem 2.8, we know

$$D(G) \leqslant P_{1,n}(1 + \log(a_1) + \log(a_1a_2) + \cdots + \log(a_1a_2 \ldots a_{n-1})).$$

We use the facts that $P_{1,n-1} \mid D(G)$ and $(P_{1,n-1} + P_{1,n} - 1) \mid D(G)$. These two factors are relatively prime, so $P_{1,n-1}(P_{1,n-1} + P_{1,n} - 1) \mid D(G)$, and therefore,

$$P_{1,n-1}(P_{1,n-1}+P_{1,n}-1) \leqslant D(G) \leqslant P_{1,n}(1+\log(a_1)+\log(a_1a_2)+\cdots+\log(a_1a_2\ldots a_{n-1})).$$

Simplifying, we get

$$P_{1,n-1} + P_{1,n} - 1 \leqslant a_n(1 + \log(a_1) + \log(a_1a_2) + \cdots + \log(a_1a_2 \ldots a_{n-1})).$$

The condition that the invariant factors are distinct is equivalent to $a_i \geqslant 2$ for each i. We claim this inequality doesn't hold for $n \geqslant 2$. To prove this claim, we use induction on rank. The base case is $n = 2$. The inequality becomes

$$a(a + ab - 1) \leqslant b(1 + \log a).$$

However, note that $a^2 > 1 + \log a$ for all $a \geqslant 2$, so $a(a-1) + a^2b \geqslant a^2b > b(1 + \log a)$, which means the inequality does not hold, as desired.

Assume that this inequality holds for some values of $a_1, a_2, \ldots, a_n$ with $n \geqslant 3$. We claim it holds when you decrease a_1 by 1. If a_1 is decreased by 1, the left hand side decreases by $P_{2,n-1} + P_{2,n}$ and the right hand side decreases by $a_n(n-1)\log\frac{a_1}{a_1-1} \leqslant a_n(n-1)\log 2$. But $P_{2,n} = P_{2,n-1}a_n \geqslant 2^{n-2}a_n > a_n(n-1)\log 2$ if $n \geqslant 3$, so the left hand side decreases more than the right hand side which means the inequality still holds.

Now assume the inequality does not hold for any rank n group. We claim that it does not hold for any rank $n+1$ group. Assume for the sake of contradiction that the inequality held for the sequence $a_1, a_2, \ldots, a_{n+1}$. Then applying the previous claim, we can continuously reduce a_1 to 1 and the inequality still holds. But if $a_1 = 1$, we are left with a group of rank n, since an invariant factor of 1 can be excluded. By the assumption that the inequality does not hold for rank n groups, the inequality does not hold for $1, a_2, \ldots, a_n, a_{n+1}$, which is a contradiction.

Applying this inductively gives the result. □

The work in this section motivates the following conjecture.

Conjecture 5.8. A group G has property P if and only if it is cyclic.

6. The Inverse Davenport Problem

In addition to determining the properties of the Davenport constant, we also investigate the inverse Davenport problem–the study of the structure of maximal 0-sequences of length $D(G)$. Information on these sequences uncovers information on the structure of the groups.

Consider two maximal 0-sequences of length $D(G)$ in group G. Denote these sequences as $\{x_i\}_{1\leqslant i\leqslant D(G)}$ and $\{y_i\}_{1\leqslant i\leqslant D(G)}$. We develop and investigate three different equivalence relations for such sequences.

Definition 6.1 (Condition 1)**.** Define $\{x_i\} \sim \{y_i\}$ if there exists a $\varphi \in \mathrm{Aut}(G)$ and a permutation $\sigma \in S_{D(G)}$ such that $\varphi(x_i) = y_{\sigma(i)}$ for each $1 \leqslant i \leqslant D(G)$.

Definition 6.2 (Condition 2)**.** Define $\{x_i\} \approx \{y_i\}$ if there exists a permutation $\sigma \in S_{D(G)}$ and $D(G)$ automorphisms, call them $\varphi_1, \varphi_2, \ldots, \varphi_{D(G)}$, such that $\varphi_i(x_i) = y_{\sigma(i)}$.

Definition 6.3 (Condition 3)**.** Define $\{x_i\} \approxeq \{y_i\}$ if there exists a permutation $\sigma \in S_{D(G)}$ such that $|x_i| = |y_{\sigma(i)}|$ for each $1 \leqslant i \leqslant D(G)$.

It is easy to see that Condition 1 is the strongest, followed by Condition 2, and finally Condition 3. Symbolically,

$$\{x_i\} \sim \{y_i\} \implies \{x_i\} \approx \{y_i\} \implies \{x_i\} \approxeq \{y_i\}.$$

Proposition 6.4. Let $\varphi : G \to H$ be a group homomorphism between finite abelian groups G and H. If $\{x_1, x_2, \ldots, x_n\}$ is a 0-sequence of G, then $\{\varphi(x_1), \varphi(x_2), \ldots, \varphi(x_n)\}$ is a 0-sequence of H.

Proof. We have $\varphi(x_1) + \varphi(x_2) + \cdots + \varphi(x_n) = \varphi(x_1 + x_2 + \cdots + x_n) = \varphi(0) = 0$. □

Remark 6.5. The converse of the above statement is not true. Consider a homomorphism $\varphi : C_4 \to C_2 \oplus C_2$ defined by $\varphi(1) = (1, 0)$. Notice that $\{\varphi(1), \varphi(1)\}$ is a 0-sequence of $C_2 \oplus C_2$ but $\{1, 1\}$ is not a 0-sequence of C_4.

However, the converse holds if $\ker(\varphi) = \{0\}$. In this case, $\varphi(x_1)+\varphi(x_2)+\cdots+\varphi(x_n) = 0$ implies that $\varphi(x_1 + x_2 + \cdots + x_n) = 0$ which in turn implies that $x_1 + x_2 + \cdots + x_n = 0$.

A natural first question to ask is if any two maximal 0-sequences are automorphic images of each other.

Question 6.6. If $\{x_i\}$ and $\{y_i\}$ are two maximal 0-sequences of length $D(G)$, is $\{x_i\} \sim \{y_i\}$?

The answer to the above question is no. We construct a counterexample. Consider the group $G \cong C_2 \oplus C_6$. From [14], $D(G) = 2 + 6 - 1 = 7$. Consider the sequences $\{x_i\}$ and $\{y_i\}$ given by

$$\{x_i\} = \{(0, 1), (0, 1), (0, 1), (0, 1), (0, 1), (1, 0), (1, 1)\}$$
$$\{y_i\} = \{(0, 1), (0, 1), (0, 1), (1, 1), (1, 1), (1, 1), (1, 0)\}.$$

Both $\{x_i\}$ and $\{y_i\}$ are 0-sequences of length 7. Furthermore neither $\{x_i\}$ nor $\{y_i\}$ have any proper 0-subsequence. The element $(0, 1)$ appears fives times in $\{x_i\}$ and there is no element of $\{y_i\}$ appearing five times, so $\{x_i\}$ and $\{y_i\}$ do not satisfy Condition 1.

In fact, the structural dissimilarity in maximal 0-sequences can go much further than this.

Question 6.7. If $\{x_i\}$ and $\{y_i\}$ are two maximal 0-sequences of length $D(G)$, is $\{x_i\} \approx \{y_i\}$?

The answer to this question is no; it is not necessary that the orders of all elements in two maximal 0-sequences are the same up to permutation. Consider this counterexample in $C_2 \oplus C_2 \oplus C_4$ with $D(G) = 6$:

$$\{x_i\} = \{(1, 1, 1), (1, 0, 0), (0, 1, 0), (0, 0, 1), (0, 0, 1), (0, 0, 1)\}$$
$$\{y_i\} = \{(0, 0, 1), (0, 0, 1), (0, 0, 1), (1, 0, 1), (0, 1, 1), (1, 1, 3)\}$$

where the order sequences are 4, 2, 2, 4, 4, 4, and 4, 4, 4, 4, 4, 4, respectively. These are clearly not permutations of one another.

In order to get a better understanding of which maximal sequences of length $D(G)$ are equivalent under the three equivalence relations, a condition on when two elements of finite abelian groups are automorphic images of one another would be helpful. However, to the best of our knowledge, there isn't an elementary condition for this in the literature. This is the motivation to explore the question, which we address in the following section.

7. Condition for Automorphic Equivalence

For the rest of the paper, let G denote a finite abelian group.

Theorem 7.1. Given x, $y \in G$, there exists $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$ if and only if $G/\langle x\rangle \cong G/\langle y\rangle$.

Lemma 7.2. (Lemma 2.1 in [11]) If H and K are finite groups with relatively prime orders,

$$\mathrm{Aut}(H) \oplus \mathrm{Aut}(K) \cong \mathrm{Aut}(H \oplus K).$$

Proposition 7.3. In order to prove that $G/\langle x\rangle \cong G/\langle y\rangle$ implies there exists $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$, it is sufficient to only consider p-groups.

Proof. We know that any additively defined finite abelian group is isomorphic to the direct sum of its Sylow p-subgroups. Therefore, if G is a finite abelian group such that $|G| = {p_1}^{e_1}{p_2}^{e^2} \dots {p_n}^{e_n}$ where each p_i is a distinct prime, then

$$G \cong \bigoplus_{i=1}^{n} H_{p_i},$$

where H_{p_j} is the unique Sylow p_j-subgroup. We will represent elements of G as an n-tuple with the i^{th} member of the tuple being an element of H_{p_i}. For $x, y \in G$, let $x = (x_{p_1}, x_{p_2}, \dots, x_{p_n})$ and $y = (y_{p_1}, y_{p_2}, \dots, y_{p_n})$, where $x_{p_i}, y_{p_i} \in H_{p_i}$ for each $1 \leqslant i \leqslant n$. We will use (x_{p_i}) to denote the element of the G (written in its Sylow p-group decomposition) with x_{p_i} as the i^{th} component and identity 0 everywhere else.

For $1 \leqslant i \leqslant n$, the order of (x_{p_i}) is a power of p_i and H_{p_i} is the only component in this decomposition of G that has order divisible by p_i, so $G/\langle x\rangle \cong \bigoplus_{i=1}^{n} H_{p_i}/\langle (x_{p_i})\rangle$ and $G/\langle y\rangle \cong \bigoplus_{i=1}^{n} H_{p_i}/\langle (y_{p_i})\rangle$. We conclude that $G/\langle x\rangle \cong G/\langle y\rangle$ is equivalent to

$$H_{p_i}/\langle (x_{p_i})\rangle \cong H_{p_i}/\langle (y_{p_i})\rangle \text{ for all } 1 \leqslant i \leqslant n.$$

Combining this with Lemma 7.2, because the orders of $H_{p_i}/\langle (x_{p_i})\rangle$ are pairwise coprime for $1 \leqslant i \leqslant n$, we see that to prove the sufficient condition for Theorem 7.1, it is sufficient to prove the condition for p-groups. □

Proof of Theorem 7.1. First, we prove that $G/\langle x\rangle \cong G/\langle y\rangle$ is a necessary condition for there to exist $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$.

Define $X = \langle x\rangle$, $Y = \langle y\rangle$. If $\varphi(x) = y$ where $\varphi \in \mathrm{Aut}(G)$, we claim that $\psi : a + X \mapsto \varphi(a) + Y$ is an isomorphism. First, we show that ψ is well-defined. Assume $a + X = b + X$, then $a - b \in X$. Hence, $\varphi(a - b) = \varphi(a) - \varphi(b) \in Y$ implies that $\varphi(a) + Y = \varphi(b) + Y$ which proves $\psi(a+X) = \psi(b+X)$. Thus, ψ is well-defined. The surjectivity of ψ follows from the fact that φ is surjective. Since x and y have the same order, the number of cosets of X in G is equal to the number of cosets of Y, so injectivity follows from surjectivity and the fact that $|G|$ is finite. Finally, ψ is a homomorphism as $\psi((a+X)+(b+X)) = \psi((a+b)+X) = \varphi(a+b) + Y = (\varphi(a) + \varphi(b)) + Y = (\varphi(a) + Y) + (\varphi(b) + Y) = \psi(a + X) + \psi(b + X)$. This implies that $G/\langle x\rangle \cong G/\langle y\rangle$.

Now, we prove that $G/\langle x\rangle \cong G/\langle y\rangle$ is sufficient.

Let G be a finite abelian p-group such that

$$G \cong \bigoplus_{i=1}^{n} C_{p^{e_i}}$$

where p is prime and $1 < e_1 \leqslant e_2 \leqslant \ldots \leqslant e_n$. From [3] we know that if $G/A \cong G/B$ where A and B are cyclic groups, there exists a $\varphi_1 \in \mathrm{Aut}(G)$ that maps the elements of A to the elements of B. Taking $A = \langle x\rangle$ and $B = \langle y\rangle$, we have

$$\varphi_1(x) = ky \text{ for some } k \in \mathbb{N}.$$

Therefore, we have that $|ky| = |x| = |y|$.

We will first prove that for the non-trivial case when x and y are not the identity elements of G, we have $\gcd(k, p) = 1$. Assume for the sake of contradiction that $p \mid k$. Since G is a p-group, we know $p \mid |y|$, so

$$0 = \frac{k}{p}(|y|y) = \frac{|y|}{p}(ky)$$

which implies $|ky| \mid \frac{|y|}{p}$ so $|ky| \neq |y|$, which is a contradiction. Therefore, $\gcd(k, p) = 1$.

From the above, we know that k^{-1} exists modulo p^m for all $m \in \mathbb{N}$. In other words, for each p^{e_i}, there exists some $a_i \in \mathbb{N}$ such that $a_i k \equiv 1 \pmod{p^{e_i}}$.

Consider $x = (x_1, x_2, \ldots, x_n)$. We can show that the map defined by $\varphi_2 : G \to G$ with $\varphi_2(x) = (a_1 x_1, \ldots, a_n x_n)$ is an automorphism. Since φ_2 is a linear map, it is also a homomorphism. If $\varphi_2(g) = \varphi_2(h)$, then $k\varphi_2(g) = k\varphi_2(h)$, implying $\varphi_2(kg) = \varphi_2(kh)$, which implies $g = h$. Therefore, φ_2 is injective. Surjectivity follows from injectivity since G is finite.

From the above, we get that φ_2 is an automorphism so $\varphi_2 \circ \varphi_1 \in \mathrm{Aut}(G)$. Then,

$$(\varphi_1 \circ \varphi_2)(x) = \varphi_2(\varphi_1(x)) = \varphi_2(ky) = y.$$

Therefore, there exists an automorphism mapping x to y, as desired. □

Theorem 7.1 establishes the necessary and sufficient conditions for two elements in a finite abelian group to be automorphic images of one another. In addition to algorithmic applications, which we discuss in Section 8, this theorem can be applied alongside other results in group theory to create corollaries that would otherwise be difficult to see.

Theorem 7.4. If x and y are both of maximal order in G, they are automorphic images of one another.

Proof. Let $G \cong C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_k}$ be a finite abelian group written in its invariant factor decomposition. Let C_{m_r}, $C_{m_{r+1}}$, …, C_{m_k} be the invariant factors of maximal order. For an element $x \in G$ to be of maximal order, there exists i where $r \leqslant i \leqslant k$ such that the ith component of x is a generator of C_{m_i}. Define this generator as x_i. We construct an automorphism φ_x composing the automorphism switching the i^{th} and the k^{th} component (which is an automorphism because C_{m_i} and C_{m_k} have the same order, the maximal order) with the automorphism mapping x_i to 1 in the k^{th} component but fixing all other components. Define $x' = \varphi_x(x)$; x' must be of the form $x' = (x'_1, x'_2, \ldots, x'_{n-1}, 1)$. We construct another map $\psi_x : G \to G$ as follows:

$$\begin{aligned}
\psi_x : (1, 0, \ldots, 0, 0) &\mapsto (1, 0, \ldots, 0, 0) \\
\psi_x : (0, 1, \ldots, 0, 0) &\mapsto (0, 1, \ldots, 0, 0) \\
&\vdots \\
\psi_x : (0, 0, \ldots, 1, 0) &\mapsto (0, 0, \ldots, 1, 0) \\
\psi_x : (x'_1, x'_2, \ldots, x'_{n-1}, 1) &\mapsto (0, 0, \ldots, 0, 1).
\end{aligned}$$

We claim ψ_x is an automorphism. First of all, ψ_x is a homomorphism by construction because the preimages of all the maps we defined is a minimal spanning set of G. Furthermore, ψ_x is surjective because the images of all the defined maps form a minimal spanning set of G. Since G is finite, injectivity is implied, so ψ_x must be an automorphism.

Similarly, φ_y can be defined mapping y to y' and ψ_y can be defined mapping y' to $(0, 0, \ldots, 0, 1)$. An automorphism mapping x to y is therefore $\varphi_y^{-1} \circ \psi_y^{-1} \circ \psi_x \circ \varphi_x$. □

The above theorem combined with Theorem 7.1 implies a result that is not immediately clear.

Corollary 7.5. Given two elements x, $y \in G$ of maximal order, $G/\langle x \rangle \cong G/\langle y \rangle$.

8. Computing Automorphic Equivalence of Two Elements

8.1. **First Algorithm: Directly Applying Smith Normal Form.** It is well known that given finite abelian group G and $x \in G$, $G/\langle x \rangle$ can be computed by Smith Normal Form [2]. Let $G \cong C_{m_1} \oplus C_{m_2} \oplus \cdots \oplus C_{m_k}$ and $x = (x_1, x_2, \ldots, x_k)$. Then, $G/\langle x \rangle$ can be computed by writing the matrix

$$\begin{bmatrix}
x_1 & x_2 & \ldots & x_k \\
m_1 & & & \\
 & m_2 & & \\
 & & \ddots & \\
 & & & m_k
\end{bmatrix}$$

in Smith Normal Form.

The time complexity to compute Smith Normal Form for integer matrices in $\mathbb{Z}(n \times m)$ is $O(n^{\theta-1}mM(n\log(||A||)))$ where $||A|| = \max\{A(i,j) \mid 1 \leqslant i,j \leqslant n\}$, $M(t)$ bounds the cost of multiplying two t-bit integers, and θ is the exponent of multiplication of two $n \times n$ matrices [18].

The most commonly used fast matrix multiplication algorithm is Strassen's algorithm with a time complexity of $O(n^{2.8074})$ ([19]). For a group of rank n, the associated matrix to describe in Smith Normal Form is an $(n+1) \times n$ matrix. Assuming multiplication is a constant time operation, this gives a time complexity of verifying $G/\langle x\rangle \cong G/\langle y\rangle$ as $O(n^{2.8074})$ where n is the rank of the group. Therefore, the problem of determining whether two elements are automorphic images of one another can be solved with a time complexity of $O(n^{2.8074})$.

8.2. **Second Algorithm: Splitting into p-groups.** We present another algorithm to compute $G/\langle x\rangle$.

To do this, we describe a simpler Smith Normal Form algorithm to compute $G/\langle x\rangle$ for p-groups which runs the Smith Normal Form algorithm in [2] in terms of ν_p.

Algorithm 8.1. Suppose that $G \cong C_{p^{e_1}} \oplus \cdots \oplus C_{p^{e_k}}$ and $x = (a_1p^{f_1}, a_2p^{f_2}, \ldots, a_kp^{f_k})$ where $p \nmid a_i$ for $1 \leqslant i \leqslant k$ if $a_i \neq 0$. If any of the a_i are equal to 0, we can simply remove the zero and remove the component of G that a_i is in. Without loss of generality, let the first component of x be zero, so $x = (0, a_2p^{f_2}, \ldots, a_kp^{f_k})$. Consider $x' = (a_2p^{f_2}, \ldots, a_kp^{f_k})$ and $G' \cong C_{p^{e_2}} \oplus \cdots \oplus C_{p^{e_k}}$, and note that $G/\langle x\rangle$ is isomorphic to $C_{p^{e_1}} \oplus G'/\langle x'\rangle$. Therefore, assume $a_i \neq 0$ for all i.

Without loss of generality, let $f_1 \leqslant f_2 \leqslant \ldots \leqslant f_k$ (so the invariant factors of G do not need to be from least to greatest). We only need to consider $x = (p^{f_1}, p^{f_2}, \ldots, p^{f_k})$ since it is an automorphic image of $(a_1p^{f_1}, a_2p^{f_2}, \ldots, a_kp^{f_k})$.

First, we write down the following list:

$$\begin{array}{cccc} f_1 & f_2 & \ldots & f_k \\ e_1 & e_2 & \ldots & e_k. \end{array}$$

Let a_{mn} denote the element of that list in the mth row and the nth column. The algorithm is as follows. For each $1 \leqslant i \leqslant k-1$, add $\max(0, a_{2i} - a_{1i})$ to all a_{1j} where $i+1 \leqslant j \leqslant k$ and then erase the larger value among a_{2i} and a_{1i}. Finally, erase the larger value among a_{2k} and a_{1k}. There is one value left per column, which are the powers of the invariant factors of the quotient group.

Example 8.2. As an example, we compute $(C_2 \oplus C_4 \oplus C_8 \oplus C_8)/\langle(2,1,2,4)\rangle$. Here, $f_1 = 0$, $f_2 = f_3 = 1$, $f_4 = 2$, $e_1 = 2$, $e_2 = 1$, $e_3 = e_4 = 3$. Our list is

$$\begin{array}{cccc} 0 & 1 & 1 & 2 \\ 2 & 1 & 3 & 3 \end{array}.$$

Proceeding with the algorithm,

$$\begin{array}{ccccccccccccccc} 0 & 3 & 3 & 4 & & 0 & & 3 & 4 & & 0 & & 3 & 4 & & 0 & & 3 & \\ & 1 & 3 & 3 & \to & & 1 & 3 & 3 & \to & & 1 & & 3 & \to & & 1 & & 3 \end{array}$$

where each stage is the operation run on the succeeding column, so our quotient group is $C_{2^0} \oplus C_{2^1} \oplus C_{2^3} \oplus C_{2^3}$ which is what we would expect if we ran the normal Smith Normal Form algorithm.

Now, we describe the algorithm for computing $G/\langle x \rangle$. We break the algorithm into two steps: first, we decompose G into the direct sum of its Sylow p-subgroups, and second, we compute $G/\langle x \rangle$ when G is a p-group.

Let G be a finite abelian group such that

$$G \cong C_{a_1} \oplus C_{a_2} \oplus \cdots \oplus C_{a_n} \cong \bigoplus_{i=1}^{t} H_{p_i},$$

where the first representation is its invariant factor decomposition and the H_{p_i} are Sylow p-subgroups with distinct primes.

Decomposing G into a product of p-groups requires finding the prime factorization of a_n.

Several algorithms exist to do this; we can use the general number field sieve (see [5]), which has heuristic time complexity

$$O(\exp(((64/9)^{1/3} + o(1))(\log a_n)^{1/3}(\log \log a_n)^{2/3})).$$

Let $d(k)$ be the number of prime factors of k (not necessarily distinct). Clearly,

$$d(a_1) \leqslant d(a_2) \leqslant \cdots \leqslant d(a_n) \leqslant \log_2(a_n).$$

For each distinct prime factor p of a_n, we can find $\nu_p(a_i)$ for each i, which takes

$$O(d(a_1) + d(a_2) + \cdots + d(a_n)) = O(nd(a_n))$$

time overall. This gives us our decomposition into p-groups as for each prime p, the corresponding p-group is

$$\bigoplus_{i=1}^{n} C_{p^{\nu_p(a_i)}}.$$

The time complexity of this step is $O(nd(a_n))$. We can also do this for x and compute $\nu_p(x_i)$ for each $p \mid a_n$ and $1 \leqslant i \leqslant n$, which we will use later. This is also $O(nd(a_n))$ since we can perform the exact same algorithm.

Due to Lemma 7.2, it is sufficient to find the quotient of each p-group component of G by its corresponding p-group component of $\langle x \rangle$. Since this can be done independently for each prime, we instead describe an algorithm to compute $H/\langle x \rangle$ when H is a p-group that runs in $O(k \log k)$ time, where k is the rank of H.

Algorithm 8.1 requires the sequence f to be sorted from least to greatest. We can sort f in $O(k \log k)$ and move around the respective elements in e. The remaining algorithm involves the following procedure:

- Add some integer to the rest of the elements in the array f;
- Find the value of f at any position in the list;
- Find the value of e at any position in the list.

Notice that querying for a value in e is $O(1)$, as the list is always constant. While doing range add queries on arbitrary intervals and querying a point can be done in $O(k \log k)$ using a Segment Tree, for this specific use case, we can do it in $O(k)$ since we specifically do range add queries on suffixes. We create a variable, call it **sum**, initialized at 0 storing the amount we need to add to the rest of the array. At each index of f, we add **sum** to

the value at f. After processing that specific index i, we can calculate $\max(0, e_i - f_i)$ and add this value to **sum** since this is the value we are adding to the rest of the array f.

To summarize, given a group G with rank n, it takes us $O(nd(a_n))$ to decompose it into its p-group components after prime factorizing. Each p-group component has rank at most n and therefore it takes worst case $O(n \log n)$ to find the quotient group for a p-group. The number of p-group components is $d(a_n)$, so the complexity of computing the quotient groups is $O(n \log(n) d(a_n))$. We can replace all the $d(a_n)$ terms with $\log a_n$, since $d(a_n) \leqslant \log_2(a_n)$. Adding all the terms together, we get a complexity of

$$O(\exp(((64/9)^{1/3} + o(1))(\log a_n)^{1/3}(\log \log a_n)^{2/3}) + n \log n \log a_n),$$

which is verified in Appendix A.

The algorithm is most feasible when $a_n \leq 10^{20}$ due to the large complexity contributed by prime factorizing the exponent.

Now, we compare our two algorithms described above. Although the complexity of the algorithm in Section 8.2 is a significant improvement from our algorithm in Section 8.1, it is helpful to know the rank at which the former outerforms the latter, since the latter has a large constant factor. The number of operations required assuming the exponent of the group is 10^{20} can be approximated with

$$2 \cdot 10^7 + 4n \cdot 67 + 67n \log n,$$

where the first quantity is from the prime factorization of a_n, the second is from computing ν_p and running the algorithm, and the third is from the sorting.

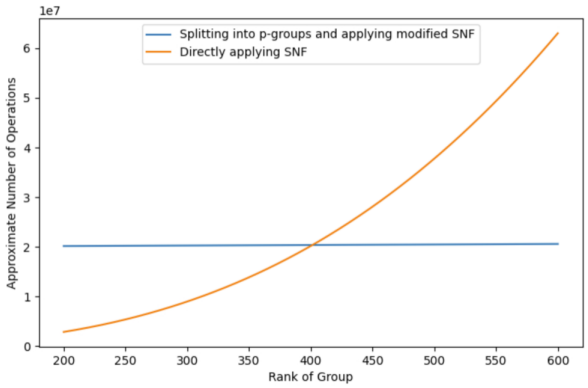

FIGURE 1. Comparing the speed of directly applying SNF versus splitting G into p-groups and applying modified SNF on each, assuming the exponent is 10^{20}.

Directly applying Smith Normal Form on $G/\langle x \rangle$ (the algorithm in Section 8.1) is faster for groups of rank below (approximately) 400 assuming the exponent of G is 10^{20}, but for groups of larger ranks, our second algorithm (the algorithm in Section 8.2) is optimal. When the exponent of G is small, our algorithm is faster at much smaller ranks.

8.3. **Comparing Algorithms.** To the best of our knowledge, there is no literature on the time complexity of an algorithm that computes whether two elements of a group are automorphic images of one another. Theorem 7.1 combined with efficient algorithms to determine whether two quotient groups are isomorphic provides a strategy to implement

an efficient algorithm for this purpose. It is more reasonable to compute the Smith Normal Form for integer matrices to check if $G/\langle x\rangle \cong G/\langle y\rangle$ than determine an automorphism that maps x to y.

Naively, a brute force algorithm to determine if x and y are automorphic images would be to compute $\mathrm{Aut}(G)$ and then iterate through it, computing $\varphi(x)$ for all $\varphi \in \mathrm{Aut}(G)$ and checking if it is equal to y. In the worst-case scenario, where x and y are not automorphic images of each other, we must iterate through the entirety of $\mathrm{Aut}(G)$. The best known algorithm to compute $\mathrm{Aut}(G)$ implemented in GAP is described by Eick, Leedham-Green and O'Brien. The time complexity of this algorithm is dominated by n^7 where n is the rank of G (see Section 11.1 in [8]). If we run this algorithm and then iterate through all elements of $\mathrm{Aut}(G)$ to check if $\varphi(x) = y$, we obtain a time complexity of $O(|\mathrm{Aut}(G)|+n^7)$. Our algorithm compares very favorably at $O(n^{2.8})$. Furthermore, we can show that $\mathrm{Aut}(G)$ is exponential in n, where n is the rank of G.

Proposition 8.3. $|\mathrm{Aut}(G)|$ is exponential in n, where n is the rank of G.

Proof. From Lemma 7.2, it suffices to assume that G is a p-group. Let $G \cong \bigoplus_{i=1}^{k} C_{p^{m_i}}^{n_i}$ and let $n = n_1 + n_2 + \cdots + n_k$ be the rank of G. A corollary of Theorem 4.1 in [11] is $|\mathrm{Aut}(C_{p^{m_i}}^{n_i})| = p^{(m_i-1)n_i{}^2} \prod_{j=0}^{n_i-1} (p^{n_i} - p^j) = p^{m_i n_i{}^2} \prod_{j=0}^{n_i-1} \left(1 - \frac{1}{p^{n_i-j}}\right)$. Since $p \geqslant 2$, this gives $|\mathrm{Aut}(C_{p^{m_i}}^{n_i})| \geqslant \frac{p^{m_i n_i{}^2}}{2^{n_i}}$. For $p = 2$, we have $|\mathrm{Aut}(C_{p^{m_i}}^{n_i})| \geqslant \frac{p^{m_i n_i{}^2}}{2^{n_i}} = 2^{n_i(m_i n_i - 1)} > 2^{n_i}$. For $p > 3$, $|\mathrm{Aut}(C_{p^{m_i}}^{n_i})| \geqslant \frac{p^{m_i n_i{}^2}}{2^{n_i}} > \left(\frac{p}{2}\right)^{n_i}$. Considering elements of G to be k-tuples, we can construct an automorphism in G by individual component-wise automorphisms of $C_{p^{m_i}}^{n_i}$. This gives $|\mathrm{Aut}(G)| \geqslant \prod_{i=1}^{k} |\mathrm{Aut}(C_{p^{m_i}}^{n_i})|$. For $p = 2$, this implies that $|\mathrm{Aut}(G)| > \prod_{i=1}^{k} 2^{n_i} = 2^n$. For $p > 2$, this implies that $|\mathrm{Aut}(G)| \geqslant \prod_{i=1}^{k} \left(\frac{p}{2}\right)^{n_i} = \left(\frac{p}{2}\right)^{n}$. In either case, we prove that $|\mathrm{Aut}(G)|$ is exponential in the rank of G. □

Since we have shown that $|\mathrm{Aut}(G)|$ is exponential in the rank n, the runtime of the naive algorithm, which has time complexity of $O(|\mathrm{Aut}(G)| + n^7)$, is exponential in n. This is significantly worse than our runtime of $O(n \log(n) \log a_n)$, especially for groups with small exponent. See Appendix A for an implementation of Algorithm 8.1 and its numerical runtime analysis.

9. Computing Automorphic Orbits

Consider a finite abelian group G with invariant factor decomposition $C_{d_1} \oplus C_{d_2} \oplus \cdots \oplus C_{d_n}$. We describe an algorithm that computes the automorphic orbits in G with a time complexity of $O(\sqrt{|G|}2^n n \log n)$. However, this algorithm works significantly faster for most groups G and achieves this complexity only when the invariant factors of G are relatively small.

Previous works computed the number of orbits in finite abelian groups (see [7]). We compute the orbits (hence also the number of orbits) by reducing each element to a specific form and applying Algorithm Two to compute the quotient group, which can be used to determine which orbit each element fits into.

Define the orbits of G as the equivalence classes by $\sim$, where $x \sim y$ if and only if there exists $\varphi \in \mathrm{Aut}(G)$ such that $\varphi(x) = y$. Equivalently, we can define them as the orbits of the natural action of $\mathrm{Aut}(G)$ on G.

By the same process as described earlier (in the algorithm in Section 8.2), we can split G into a direct sum of p-groups. In this case, we can ignore the complexity of prime factorizing a_n, since it can be done in much faster than $\sqrt{|G|}$ time, and therefore, this step is insignificant compared to the overall complexity.

We first describe the algorithm for finding automorphic orbits in p-groups and show how these orbits can subsequently be combined to find the orbits of G.

Let H be a p-group of the form $\bigoplus_{i=1}^{n} C_{p^{e_i}}$. For each pair of elements x, y in the orbit $\mathcal{O}$ of H, recall that $H/\langle x\rangle \cong H/\langle y\rangle$ by Theorem 7.1. Therefore, we can say a group K *corresponds* to orbit $\mathcal{O}$ if $H/\langle x\rangle \cong K$ for each $x \in \mathcal{O}$.

Consider an element $x \in H$ of the form $(a_1 p^{b_1}, a_2 p^{b_2}, \cdots, a_n p^{b_n})$, where $0 \leqslant b_i \leqslant e_i$ and $p \nmid a_i$ for each $1 \leqslant i \leqslant n$. The element x is in the same orbit as $(p^{b_1}, p^{b_2}, \cdots, p^{b_n})$ because there exists an automorphism between x and $(p^{b_1}, p^{b_2}, \cdots, p^{b_n})$ component-wise for each component. We say $(p^{b_1}, p^{b_2}, \cdots, p^{b_n})$ is the *reduced form* of x. Therefore, it is sufficient to compute the orbits of the elements that satisfy $a_1 = a_2 = \cdots = a_n = 1$.

There are $\prod_{i=1}^{n}(e_i+1)$ such elements. Therefore, for each element $x = (p^{b_1}, p^{b_2}, \cdots, p^{b_n})$, we can compute $H/\langle x\rangle$, and any element that has a reduced form equal to x is in the orbit corresponding to $H/\langle x\rangle$. In fact, we can compute the number of elements that have this property. Since this is independent for each component, we can show how to calculate it for the ith component and multiply this across.

If $b_i = e_i$, then there is only one distinct x as the component is just 0. If $b_i < e_i$, then we claim there are $p^{e_i - b_i} - p^{e_i - b_i - 1}$. It is sufficient to compute how many $1 \leqslant a_i \leqslant p^{e_i - b_i}$ exist such that $\gcd(a_i, p) = 1$, but this is simply $\varphi(p^{e_i - b_i})$, where φ is the Euler Totient function.

Overall, this algorithm takes $O(\prod(e_i + 1) \cdot n \log n)$ time and it computes the size of each orbit and the reduced forms that are part of that orbit.

Doing this for all k different p-groups gives k sets of orbits $S_1, S_2, \ldots, S_k$. The number of orbits is $|S_1||S_2| \cdots |S_k|$, since we choose orbits $\mathcal{O}_1 \in S_1, \mathcal{O}_2 \in S_2, \cdots, \mathcal{O}_k \in S_k$ and each of these k-tuples of orbits corresponds to a unique orbit in G.

We can also find the properties of this unique orbit. The size of the orbit is the product of the sizes of the k individual orbits. The representative elements of the orbit are any combination of k representative elements, one from each $\mathcal{O}_i$. In other words, $x \in G$ is a representative element for this orbit if and only if the component of x for the ith p-group is one of the representatives for $\mathcal{O}_i$.

Overall, the time complexity is

$$O\left(\prod_{\substack{p \leqslant n \\ p \text{ prime}}} \prod_{i=1}^{n} (\nu_p(d_i) + 1) \cdot n \log n\right).$$

Notice that the double product simply computes the number of factors of d_i, so we can rewrite this as

$$O\left(\prod_{i=1}^{n} \tau(d_i) n \log n\right).$$

However, $\tau(d_i) \leqslant 2\sqrt{d_i}$, so we can bound this complexity above with

$$O\left(\prod_{i=1}^{n} (2\sqrt{d_i}) n \log n\right) = O(\sqrt{|G|} 2^n \cdot n \log n).$$

As stated earlier, when d_i are large, the $\sqrt{d_i}$ replacement is weak and the algorithm is much faster.

10. Conclusion

In this paper, we started with the Davenport constant, a purely combinatorial problem. In particular, we investigated several conjectures which led us to an interesting group theory result with many computational applications. This demonstrates the interconnectedness of mathematics: the investigation of a concept often leads to discoveries in seemingly unrelated ideas. Our work also points to several future directions for study: Conjecture 5.8 is still an open problem that can provide insights between the Davenport constant and the structure of the group. In addition, Theorem 7.1 along with the algorithms in sections 8.1 and 8.2 can help guide future investigations for applications in computational algebra. We hope to pair these algorithms with theorems in the literature to compute $\mathrm{Aut}(G)$ and other group properties.

Appendix A. Pseudocode for Algorithms and Numerical Analysis

The following is pseudocode for Algorithm 8.1, where the isAutoImage(x, y, G) function returns whether x and y are automorphic images in finite abelian group G written in its invariant factor decomposition.

```
function DecomposeElementIntoParts(G, x)
    factors ← factors(|G|)
    for p in factors do
        for i ← 1 to |G| do
            if ν_p(G[i]) ≠ 0 then
                Add i to psubindeces
            end if
        end for
        for i in psubindeces do
            Add ν_p(x[i]) to components
        end for
        Add components to pParts
    end for
    return pParts
end function
function DecomposeAbelianGroup(G)
    factors ← factors(|G|)
    for p in factors do
        for i ← 1 to |G| do
            if ν_p(G[i]) ≠ 0 then
                Add ν_p(G[i]) to components
            end if
        end for
        Add components to pParts
    end for
    return pParts
end function
function Sort(f)
    Sort f by the first element in the pair
end function
function pGroupSNF(f)
    Sort(f)
    for i ← 1 to Length(f) do
        Add addto to f[i][1]
        Add Max(0, f[i][2] − f[i][1]) to addto
    end for
    for i ← 1 to Length(f) do
        Add Min(f[i][1], f[i][2]) to final
    end for
    return final
end function
```

```
function combineLists(list1, list2)
    for i ← 1 to length(list1) do
        list[i] ← [list1[i], list2[i]]
    end for
    return list
end function
function isAutoImage(x, y, G)
    decomp ← DecomposeAbelianGroup(G)
    partsx ← DecomposeElementsIntoParts(x)
    manipx ← CombineLists(partsx, decomp)
    for i ← 1 to Length(manipx) do
        manipx[i] ← pGroupSNF(manipx[i])
    end for
    partsy ← DecomposeElementsIntoParts(y)
    manipy ← CombineLists(partsy, decomp)
    for i ← 1 to Length(manipy) do
        manipy[i] ← pGroupSNF(manipy[i])
    end for
    if manipx = manipy then
        return true
    end if
    return false
end function
```

We ran this algorithm for groups of the form C_4^n for $n \in \{3 + 10k \mid 0 \leqslant k \leqslant 16\} \cup \{2^k \mid 1 \leqslant k \leqslant 9\}$, $x = (1, 1, \dots)$, and $y = (3, 3, \dots)$ so that the exponent remains constant and small enough so that the prime factorization does not contribute significantly to the runtime. The data is shown in the table below.

Rank	2	3	4	8	13	16	23	32	33	43	53	63	64	73
Runtime (ms)	1.6	2	1.8	3.6	4	4.6	8	9.2	9	17	20	26	26.6	35

Table 1. Rank vs Runtime (ms)

Rank	83	93	103	113	123	128	133	143	153	163	256	512
Runtime (ms)	44	61	72	88	106	110.6	122	145	173	199	545.8	3263

Table 2. Rank and Runtime (ms) continued

Implementing in GAP and using Python polynomial fitting code, the best fit polynomial is $0.2129476474670508x^{1.28247480729629}$, which is near-linear and consistent with the runtime of $O(n \log n)$.

Acknowledgements

The authors would like to express their heartfelt gratitude to their mentors Professor Jim Coykendall and Jared Kettinger for their knowledge, expertise, and guidance throughout

the research process. The authors also kindly thank Dr. Felix Gotti and the PRIMES-USA research program for giving them this amazing opportunity to learn and conduct research.

References

[1] William R Alford, Andrew Granville, and Carl Pomerance. There are infinitely many carmichael numbers. *Annals of Mathematics*, 139(3):703–722, 1994.
[2] Michael Artin. *Algebra*. Birkhäuser, 1998.
[3] K. Buzasi. Invariants of pairs of finite abelian groups. *Publ. Math. Debrecen*, 28(3-4):317–326, 1981.
[4] F. Chen and S. Savchev. Long minimal zero-sum sequences in the groups $c_2^{r-1} \oplus c_{2k}$. *Integers*, 14:Paper No. A23, 29, 2014.
[5] Richard E Crandall and Carl Pomerance. *Prime numbers: a computational perspective*, volume 2. Springer, 2005.
[6] Charles Delorme, Oscar Ordaz, and Domingo Quiroz. Some remarks on davenport constant. *Discrete Mathematics*, 237(1-3):119–128, 2001.
[7] Kunal Dutta and Amritanshu Prasad. Degenerations and orbits in finite abelian groups. *Journal of Combinatorial Theory, Series A*, 118(6):1685–1694, 2011.
[8] B. Eick, C. R. Leedham-Green, and E. A. O'Brien. Constructing automorphism groups of p-groups. *Communications in Algebra*, 30(5):2271–2295, 2002.
[9] Weidong Gao, Alfred Geroldinger, and David J Grynkiewicz. Inverse zero-sum problems iii. *arXiv preprint arXiv:0801.3792*, 2008.
[10] Alfred Geroldinger and Rudolf Schneider. On davenport's constant. *Journal of Combinatorial Theory, Series A*, 61(1):147–152, 1992.
[11] Christopher J Hillar and Darren L Rhea. Automorphisms of finite abelian groups. *The American Mathematical Monthly*, 114(10):917–923, 2007.
[12] W Narkiewicz. A note on elasticity of factorizations. *Journal of Number Theory*, 1(51):46–47, 1995.
[13] John E Olson. A combinatorial problem on finite abelian groups, i. *Journal of number theory*, 1(1):8–10, 1969.
[14] John E Olson. A combinatorial problem on finite abelian groups, ii. *Journal of Number Theory*, 1(2):195–199, 1969.
[15] Kenneth Rogers. A combinatorial problem in abelian groups. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 59, pages 559–562. Cambridge University Press, 1963.
[16] Wolfgang A Schmid. Inverse zero-sum problems ii. *arXiv preprint arXiv:0801.3747*, 2008.
[17] Aleen Sheikh. *The Davenport constant of finite abelian groups*. PhD thesis, Royal Holloway, University of London, 2017.
[18] Arne Storjohann. Near optimal algorithms for computing smith normal forms of integer matrices. In *Proceedings of the 1996 international symposium on Symbolic and algebraic computation*, pages 267–274, 1996.
[19] Volker Strassen. Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354–356, 1969.
[20] Robert J Valenza. Elasticity of factorization in number fields. *Journal of Number Theory*, 36(2):212–218, 1990.
[21] P. van Emde Boas and D. Kruyswijk. A combinatorial problem on finite abelian groups iii. *Stichting Mathematisch Centrum. Afd. Zuivere Wiskunde*, 1969.

PRIMES-USA
Email address, A. Agarwal: arjunagarwal010@gmail.com
Email address, R. Chen: rachelrxchen@gmail.com
Email address, R. Garg: rohangarg2008@gmail.com