M2R Group Project

Imperial College London
Department of Mathematics

Principle of Inclusion-Exclusion and its Applications

Author: Zhenyu Chen, Clara Heng, Tyler Farghly, Jose Pablo Folch, Yidan Xu
Supervisor: Dr Lynda White *Group: 43*

Abstract

One of the most important concepts in combinatorial theory is the Principle of Inclusion-Exclusion (or *PIE*); an enumeration formula based on a strategic combination of over-generous inclusions, and compensating exclusions. This counting technique has yielded the solution to many challenging problems and motivated the formation of new branches of combinatorial theory. Moreover, the generality of the PIE allows its application to be found in fields outside of probability, especially in many areas of number theory and algebra.

This project aims to explore some of the core ideas and applications surrounding the PIE. First, we notate and prove the bare statement and discuss some basic examples in which it can be applied directly. Following that, we discuss restricted permutation problems as an application of the PIE by first introducing the topic of derangements then extending to the general case. These problems can be interpreted as placing rooks on a chessboard with constraints, hence introducing the subject of *Rook polynomials*. Specifically, we look over some well discussed board shapes and explore various algorithms for polynomial computation. On top of this, we introduce the famous *Euler's Totient function* and by characterising it as a counting function its link to the PIE can be illustrated. From this, some important properties of the function are derived and then its applications in RSA cryptography are discussed. Lastly, we introduce the *Möbius inversion* for a *partially ordered set* as a means of inverting some general summation. We then show that the PIE along with its applications can be thought of as examples of such an inversion. This section concludes with a brief discussion about how this abstraction has impacted the way in which the PIE has been applied and extended.

1 Introduction

The Principle of Inclusion-Exclusion or *PIE* is presents a straightforward method for finding the number of elements in the union of given sets, by strategically over-counting and under-counting such that each element is counted exactly once. The general form of the principle for sets $X_1, ..., X_n$ states

$$|X_1 \cup \cdots \cup X_k| = \sum_{i=1}^{k} |X_i| - \sum_{i=1}^{k}\sum_{j=1}^{k} |X_i \cap X_j| + \cdots + (-1)^{k-1}|X_1 \cap \cdots \cap X_k|.$$

The principle was first mentioned by Abraham de Moivre in 1718 but has remained a topic of intrigue due to its alluring nature and plentiful disparate applications. This report aims to provide a review of this principle and explore some of its applications with a focus on combinatorics.

In Section 2, we start by considering the cases of two and three sets and then generalise it to n sets by an inductive proof. PIE appears to be extremely useful for many interesting problems, such as finding how many permutations of a set that do not have fixed points. This is known as derangement and will be addressed in Section 3.

In the section dedicated to the problem of derangements, we will firstly develop recursive formulas for efficient computation, then the probability density function for the probability that a random permutation of a index set is a derangement. Moreover, we will develop the idea of *Partial Derangements*, which

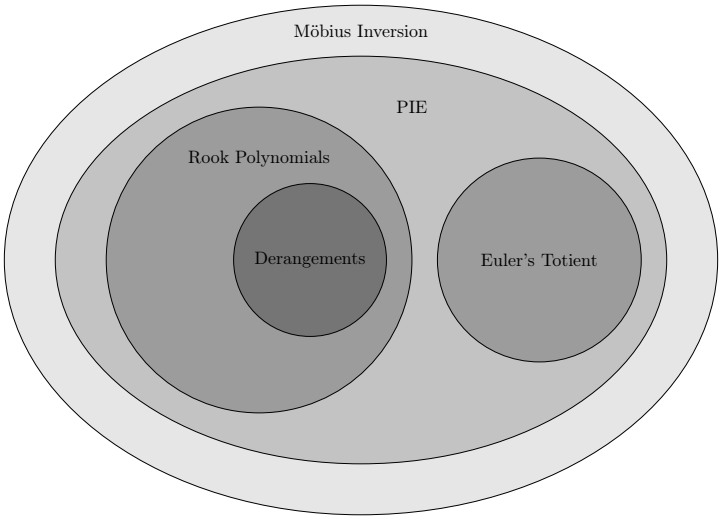


Figure 1: Project Structure in Euler's Diagram

consider permutations with exactly k fixed points. An extended and more complex example looks at the matching problem, which is widely discussed in probability, and applied in many other disciplines of science. Finally, we consider the problem of counting derangements in a different perspective, which can be nicely represented by a chessboard with darkened squares. The idea of placing non-attacking rooks on restricted positions is the building block of the theory developed in the next section.

To efficiently count the number of rook arrangements we introduce the *Rook Polynomials*, these are the generating functions for the number of ways of placing k non-capturing rooks on the darkened squares. We will show that the problem can be greatly simplified by swapping rows and columns, and introduce theorems which allow us to decompose the board into smaller, simpler boards. Motivated by the idea of optimising the decomposition process, we will investigate different algorithms of choosing cell(s) to decompose the board. We examine the *Block Decomposition Algorithm* introduced in [21], which utilises the symmetry property of some boards to ease the computation of the rook polynomial. Furthermore, we propose an *Improved Cell Decomposition Algorithm*, based on what was first developed by *John Riordan* in [24], and improved the way a cell is chosen. Finally we investigate specific cases of boards, both rectangular and staircase shape, along with associated examples.

Following in the next section, we will introduce the *Euler's Totient Function*, along with some of its properties and general formula. From this, we will further show how the understanding of the Euler's Totient function stems from the PIE. Moreover, we may then ask if there is anything we can say about the totient numbers and if there is an infinite number of integers with the same totient number? Using the general formula we have established, we can find the solutions to the above questions and further explore a common combinatorial problem, i.e. *the Necklace Problem*, and its widely known application in *RSA Cryptography*. We will further end off this section by discussing how it is interrelated to the *Classical Möbius Function*.

In the final section we explore an abstract generalisation of the over-counting under-counting procedure for any set that has some form of order. We start by defining an abstract notion of order with the partially ordered set and then define a generalised Möbius inversion formula for such a set, with its corresponding Möbius function. We then illustrate how a lot of the problems discussed in this report could be phrased as Möbius inversion problems. In particular, we express the PIE as a special case of such a problem and discuss how this fact has extended on the PIE and has been responsible for many new results in combinatorics.

Contents

2 Principle of Inclusion-Exclusion

2.1 Introduction

Let us begin with a short example to motivate the case of two sets.

Example 2.1.1. How many integers in $\{1, \cdots, 120\}$ are divisible by 4 and 6?

Solution. Firstly, we shall consider the number of integers which are divisible by just 4 and just 6. Clearly, there are $\frac{120}{4} = 30$ integers divisible by 4, and $\frac{120}{6} = 20$ integers divisible by 6. However, there are some integers that are divisible by both 4 and 6 which would have double-counted. Hence, if an integer is divisible by 4 and 6 it is divisible by its lowest common multiple which is 12. Consequently, there are $\frac{120}{12} = 10$ integers divisible by 12 and so the total number of integers in $\{1, \cdots, 120\}$ that are divisible by 4 and 6 is $30 + 20 - 10 = 40$.

Remark. The above process is similar to the idea of the Principle of Inclusion-Exclusion, where we include the cardinalities of two sets and exclude the cardinalities of intersections.

2.2 Case of Two and Three Sets

For the sake of tidiness we introduce the following notation: suppose we have n sets $A_1, ..., A_n$, then $A_1A_2...A_n$ denotes the intersection $A_1 \cap A_2 \cap ... \cap A_n$.

2.2.1 Inclusion-Exclusion for Two Sets

Let X_1 and X_2 be two finite sets, then

$$|X_1 \cup X_2| = |X_1| + |X_2| - |X_1X_2|$$

Proof. **Case 1:** If the intersection of X_1 and X_2 is the empty set, then clearly the elements of the union of X_1 and X_2 is equal to the sum of the elements of X_1 and X_2
Case 2: If the intersection of X_1 and X_2 is not the empty set, then $|X_1| + |X_2|$ count the intersection twice, so we exclude the intersection once to ensure that every element is counted only once. □

2.2.2 Inclusion-Exclusion for Three Sets

Let X_1, X_2 and X_3 be finite sets, then

$$|X_1 \cup X_2 \cup X_3| = |X_1| + |X_2| + |X_3| - |X_1X_2| - |X_1X_3| - |X_2X_3| + |X_1X_2X_3|$$

Proof. This proof is provided by Prof. Nige Ray [23]. The idea of the proof is to separate the three sets into groups of one and two sets, and then apply the Principle of Inclusion-Exclusion for two sets as shown above.

$$\begin{aligned} |(X_1 \cup X_2) \cup X_3| &= |X_1 \cup X_2| + |X_3| - |(X_1 \cup X_2)X_3| \\ &= |X_1| + |X_2| - |X_1X_2| + |X_3| - |(X_1 \cup X_2)X_3| \end{aligned} \tag{1}$$

$$|(X_1 \cup X_2)X_3| = |X_1X_3 \cup X_2X_3| = |X_1X_3| + |X_2X_3| - |X_1X_2X_3| \tag{2}$$

Combining equations (1) and (2),

$$|X_1 \cup X_2 \cup X_3| = |X_1| + |X_2| + |X_3| - |X_1X_2| - |X_1X_3| - |X_2X_3| + |X_1X_2X_3| \tag{3}$$

□

2.2.3 Inclusion-Exclusion for N sets

We will now extend the Principle of Inclusion-Exclusion to the general case of n sets by mathematical induction. As we have already proven the statement to be true for two and three sets, we have shown that $P(2)$ and $P(3)$ are true. Now, we assume that the statement is true for all $m \leq k$ sets, so $P(k)$ is the mathematical statement "$|X_1 \cup \cdots \cup X_k| = \sum_{i=1}^{k} |X_i| - \sum_{i=1}^{k}\sum_{j=1}^{k} |X_i X_j| + \cdots + (-1)^{k-1}|X_1 \cdots X_k|$".
Then for $P(k+1)$, we have the following:

$$\begin{aligned}
|X_1 \cup \cdots \cup X_{k+1}| &= |X_1 \cup \cdots \cup X_k| + |X_{k+1}| - |(\cup_{i=1}^{k} X_i) X_{k+1}| \\
&= \sum_{i=1}^{k} |X_i| - \sum_{1 \leq i,j \leq k} |X_i X_j| + \cdots + (-1)^{k-1}|X_1 \cdots X_k| + |X_{k+1}| + |\cup_{i=1}^{k} X_i X_{k+1}| \\
&= \sum_{i=1}^{k+1} |X_i| - \sum_{1 \leq i,j \leq k} |X_i X_j| + \cdots + (-1)^{k-1}|X_1 \cdots X_k| \\
&\qquad - \sum_{i=1}^{k} |X_i X_{k+1}| + \cdots + (-1)^{k-1}|X_1 \cdots X_{k+1}| \\
&= \sum_{i=1}^{k+1} |X_i| - \sum_{1 \leq i,j \leq k+1} |X_i X_j| + \sum_{1 \leq i,j,k \leq k+1} |X_i X_j X_k| + \cdots + (-1)^{k}|X_1 \cdots X_{k+1}|
\end{aligned}$$

Since the base case holds and $P(k) \implies P(k+1)$, we have shown that the statement $P(n)$, i.e. $|X_1 \cup \cdots \cup X_n| = \sum_{I \subseteq \{1,\cdots,n\}, I \neq \emptyset} (-1)^{|I|-1} |\bigcap_{i \in I} X_i|$ is true for all $n \in \mathbb{Z}_{\geq 1}$.

2.3 General Case of Inclusion-Exclusion

In this section, we will give an alternative proof of the principle.

Theorem 2.3.1 (Principle of Inclusion-Exclusion)**.** *If U is a finite set and $\{X_j\}_{j=1}^{n}$ is a collection of n subsets, then*

$$|X_1 \cup \cdots \cup X_n| = \sum_{I \subseteq \{1,\cdots,n\}, I \neq \emptyset} (-1)^{|I|-1} |\bigcap_{i \in I} X_i| \tag{4}$$

Proof. The key idea is to think of each element in $X_1 \cup \ldots \cup X_n$ individually and examine how it contributes to the sum in Equation (4). Firstly, we suppose an element $x \in X_1 \cup \ldots \cup X_n$ belongs to exactly l sets where $1 \leq l \leq n$, then we will prove that there is only 1 contribution from x. As we assume that x belongs to exactly l of the subsets, it has a total of l contributions when $|I| = 1$, where I is the index set. Next, x has a total of -$\binom{l}{2}$ contributions to the sum involving pairwise intersections where $X_i X_j$ is the intersection of X_i and X_j, $i, j = 1, \cdots, n$

$$-|X_1 X_2| - \cdots - |X_{n-1} X_n| \tag{5}$$

A similar argument shows that if $k \leq l$, then x contributes a total of

$$(-1)^{k-1}\binom{l}{k} = (-1)^{k-1}\Big(\frac{l!}{k!(l-k)!}\Big) \tag{6}$$

to the sum in Equation (4) when it involves k-fold intersections.
Finally, for $k > l$, there are no k-fold intersections that contain x. Hence, x makes no contribution to the corresponding sums in Equation (4).
Putting these observations together we see that x makes a total contribution of

$$l - \binom{l}{2} + \binom{l}{3} - \cdots + (-1)^{l-1}\binom{l}{l} \tag{7}$$

Next, by considering the following application of the Binomial Theorem,

$$\begin{aligned} l - \binom{l}{2} + \binom{l}{3} - \cdots + (-1)^{l-1}\binom{l}{l} &= 1 - \Big[1 - l + \binom{l}{2} - \binom{l}{3} + \cdots + (-1)^{l-1}\binom{l}{l}\Big] \\ &= 1 - \sum_{j=0}^{l} (-1)^j (1)^{l-j} \binom{l}{j} \\ &= 1 - (1-1)^l \\ &= 1 \end{aligned}$$

Thus, we have established that for any x which belongs to exactly l of the subsets X_j, it makes only one contribution as shown in the above equation. Lastly, as every $x \in X_1 \cup \cdots \cup X_n$, x must belong to at least one of the X_j, this then establishes the Principle of Inclusion-Exclusion.

□

2.4 Examples and Extensions

Example 2.4.1. Each box of cereal contains one of n possible cards, where the cards are randomly allocated to the boxes. You buy N boxes of cereal, where $N > n$. What is the probability that you have a complete set of cards?

Solution. We define A_i as the event that the i^{th} card is not contained in the N boxes.
Then the probability that N boxes contain a complete set of cards is:

$$\begin{aligned} P(\bar{A_1}\bar{A_2}\cdots\bar{A_n}) &= P(\overline{A_1 \cup A_2 \cdots \cup A_n}) \\ &= 1 - P(A_1 \cup A_2 \cdots \cup A_n) \end{aligned}$$

Next, we proceed to calculate the probability of the various intersections of $A_i s$ as follows:

$$\begin{aligned} P(A_i) &= \Big(\frac{n-1}{n}\Big)^N \qquad \text{for } i = 1 \to n \\ P(A_i A_j) &= \Big(\frac{n-2}{n}\Big)^N \qquad \text{for } i, j = 1 \to n, i \neq j \\ P(\underbrace{A_i A_j \cdots A_m}_{\text{k}}) &= \Big(\frac{n-k}{n}\Big)^N \qquad \text{for } k = 1 \to n \end{aligned}$$

Thus, by applying the Principle of Inclusion-Exclusion,

$$\begin{aligned} 1 - P(A_1 \cup A_2 \cdots \cup A_n) &= 1 - \sum P(A_i) + \sum P(A_i \cap A_j) - \cdots\cdots \\ &= 1 + \sum_{k=1}^{n} \binom{n}{k} (-1)^k \Big(\frac{n-k}{n}\Big)^N \end{aligned}$$

Extension. Moreover, we shall consider when each box of cereal contains l cards ($l < n$), where the cards are randomly allocated. Then, what is the probability that you have a complete set of cards after buying N boxes?
Now we consider two scenarios.

Scenario 1: Boxes can contain duplicate cards.
Let P(A_i) be the probability that all N boxes do not contain the i^{th} card.

$$\begin{aligned} P(A_i) &= \Big(\frac{n-1}{n}\Big)^{lN} \qquad \text{for } i = 1 \to n \\ P(A_i A_j) &= \Big(\frac{n-2}{n}\Big)^{lN} \qquad \text{for } i, j = 1 \to n, i \neq j \\ P(\underbrace{A_i A_j \cdots A_m}_{\text{k}}) &= \Big(\frac{n-k}{n}\Big)^{lN} \qquad \text{for } k = 1 \to n \end{aligned}$$

Hence, the following result is obtained:

$$1+\sum_{k=1}^{n}\binom{n}{k}(-1)^k\Big(\frac{n-k}{n}\Big)^{lN}$$

Scenario 2: No repeated cards contain in one box.
Let P(A_i) be the probability that all N boxes do not contain the i^{th} card.

$$\begin{aligned}
P(A_i) &= \left(\frac{(n-1)(n-2)(n-3)\cdots(n-l)}{n^l}\right)^N && \text{for } i=1\to n\\
P(\underbrace{A_iA_j\cdots A_m}_{k}) &= \left(\frac{(n-k)(n-k-1)\cdots(n-k-l+1)}{n^l}\right)^N && \text{for } k=1\to n-l\\
P(\underbrace{A_iA_j\cdots A_m}_{k}) &= 0 && \text{for } k=n-l+1\to n
\end{aligned}$$

Hence, the following result is obtained:

$$1+\sum_{k=1}^{n-l}\binom{n}{k}(-1)^k\left(\frac{(n-k)(n-k-1)\cdots(n-k-l+1)}{n^l}\right)^N$$

3 Derangements

In this section we introduce the problem of Derangements, a widely discussed topic first introduced by *Pierre Raymond de Montmort* in 1708. We can start thinking of an interesting example. Imagine that you are a mailman who needs to deliver 10 parcels to 10 different people. However, you did not read the address and delivers them off randomly. So, how many ways can you deliver the parcels such that no one gets the right parcel?

3.1 Introduction

Definition 3.1.1 (Derangement)**.** Derangements are defined as permutations with no fixed points. In other words, derangements are permutations on set such that no element of the set is sent to itself.

We denote the number of derangements of the set $\{1, 2, \cdots, n\}$ as $!n$. e.g.
$!1 = 0 :$ ( );
$!2 = 1 :$ (21);
$!3 = 2 :$ (231), (312);
$!4 = 9 :$ (2143), (2341), (2413), (3142), (3412), (3421), (4123), (4312), (4321).

3.2 Standard Results

Theorem 3.2.1 (Number of derangements)**.** *The number of derangements in S_n, is given by:*

$$!n = n! \sum_{i=0}^{n} (-1)^i \frac{1}{i!}$$

Proof. Let A_i be the set of permutations that have a fixed point on the i^{th} position. Hence, the number of derangements is given by $|\bar{A}_1 \bar{A}_2 \cdots \bar{A}_n|$. By the inclusion-exclusion principle we have that:

$$|\bar{A}_1 \bar{A}_2 \cdots \bar{A}_n| = N - S_1 + S_2 - S_3 + \cdots + (-1)^n S_n$$

Where $S_1 = \sum_i |A_i|$, $S_2 = \sum_{i<j} |A_i A_j|$, and so on. Now let x be any term of the sum S_i. Then x is the number of permutations that have a fixed point at $j_1, j_2, \cdots, j_i$, for some distinct $j_k \in \{1, 2, 3, \cdots, n\}$. As there are i positions fixed, and $n - i$ positions free in the permutation, we obtain that $x = 1 \cdot (n - i)!$. In addition, there are exactly $\binom{n}{i}$ terms in S_i. Hence

$$S_i = \binom{n}{i}(n-i)! = \frac{n!}{i!(n-i)!}(n-i)! = \frac{n!}{i!}.$$

And finally we obtain

$$!n = n! + \sum_{i=1}^{n} (-1)^i \frac{n!}{i!} = n! \sum_{i=0}^{n} (-1)^i \frac{1}{i!}.$$

□

Example 3.2.1. At a dance party there are n men and n women. In how many ways can the n men choose female partners for the first dance? If everyone has to change partners, how many ways are there for the second dance?

Solution. We start by considering the first dance. The first man has n choices, and the second man has $n - 1$ choices and so on. Thus, there are $n!$ combinations for n men choosing n female partners.
For the second dance, we may think of it as a derangement problem and the answer is $!n$.

Corollary 3.2.1.1 (Probability of a random permutation being a derangement)**.** *Given a random permutation $x \in S_n$, the probability that x is a derangement is given by*

$$\frac{!n}{n!} = \sum_{i=0}^{n} (-1)^i \frac{1}{i!} \approx \frac{1}{e}$$

Furthermore,

$$\lim_{n \to \infty} \frac{!n}{n!} = \frac{1}{e}.$$

3.3 Recursive Formula

There are several formulae for computing derangements, apart from directly applying Theorem 3.2.1. In particular, we can obtain a recursive formula to find the relationship between $!n$, $!(n-1)$ and $!(n-2)$ [12].

Lemma 3.3.1.

$$!n = (n-1)(!(n-2)+!(n-1)) \tag{8}$$

Proof. For the proof, we denote D_n as the set of derangements of $\{1, \cdots, n\}$.
Note that for $n = 1$, $!n = 0$ which is trivial as the only element in the set has to be mapped to itself. Thus, there is clearly no derangement of the set. Furthermore, note that $!1 = 0$, $!2 = 1$, as $(2\,1)$ is the only derangement of $\{1, 2\}$. Hence, we have shown that Equation (8) is true for both $n = 1$ and $n = 2$. We will now show that $!n$ is a product of $n-1$ and $!(n-2)+!(n-1)$. If we are able to partition D_n into $n-1$ subsets, where each subset has $!(n-2)+!(n-1)$ elements, hence we have obtained Equation (8).
Firstly, let R_k be the set of derangements of $\{1, \cdots, n\}$ where k is in the n^{th} position for $k = 1, \cdots, n-1$. Then

$$D_n = \cup_{j=1}^{n-1} R_j$$

Let r_k be the number of elements in R_k, where $r_1 = r_2 = \cdots = r_{n-1}$. Then

$$!n = r_1 + \cdots + r_{n-1} = (n-1)r_{n-1}.$$

Now, we need to show that $|D_{n-2} + D_{n-1}| = r_{n-1}$ is the number of derangements such that $n-1$ is in the n^{th} position.
For the second step, we shall partition the permutations in R_{n-1} into two disjoint sets. One with $!(n-1)$ elements and the other with $!(n-2)$ elements. In R_{n-1}, $n-1$ appears in the last position. If we further let n to appear in the $(n-1)^{th}$ position, then the n^{th} and $(n-1)^{th}$ entry can be removed to obtain a derangement in D_{n-2}. Hence, $|D_{n-2}|$ is the number of derangements of R_{n-1} such that $n-1$ is in the n^{th} position and n is in the $(n-1)^{th}$ position.
We can now look at the remaining derangements in R_{n-1} where n is not in the $(n-1)^{th}$ position.
Let P_n be the set of derangement where n-1 is in the n^{th} position and k is in the $(n-1)^{th}$ position for some $k \neq n, n-1$. Then, to show $|P_n| =!(n-1)$, we create a bijection from P_n to D_{n-1} as the following: Consider

$$(i_1 i_2 \cdots i_n) \in P_n, \quad i_n = n-1, \quad i_j = n \text{ for } j \in \{1, \cdots, n-2\}$$
$$(a_1 a_2 \cdots a_{n-1}) \in D_{n-1}$$

Where i_j and a_j is the elements at j^{th} position of P_n and D_{n-1} respectively.
Here, the derangements in P_n has n terms, while a derangement in D_{n-1} has $n-1$ terms. As the placement of $n-1$ does not vary (in the last position), we may remove the n^{th} term and replace $i_j = n$ with $i_j = n-1$. Thus,

$$a_l = \begin{cases} i_l & \text{if } i_l \neq \text{n}, 1 \leq l \leq \text{n - 1} \\ n-1 & \text{if } i_l = \text{n} \end{cases}$$

gives a bijection between P_n and D_{n-1}. Since P_n and D_{n-1} are finite sets, creating a bijection between the two sets necessarily implies both finite sets have the same cardinality. Hence, we have $|P_n| = |D_{n-1}| = !(n-1)$. Finally, putting both cases together, we obtain have indeed shown that $|D_{n-2} + D_{n-1}| = r_{n-1}$. □

Lemma 3.3.2.

$$!n = n(!(n-1)) + (-1)^n \quad \textit{for } n \geq 2$$

The result can be established by arguing inductively.

3.4 Partial Derangements

In this section, we will be looking at derangements in a more generalised setting, allowing k fixed elements.

Example 3.4.1 (Problème des Recontres)**.** The problem asks to find the number of permutations of size n that have exactly k fixed elements.

Solution. We will solve the problem using derangements. For any permutation, if we fix k elements, then the number of ways to arrange the remaining elements such that none are fixed is simply $!(n-k)$. As there are $\binom{n}{k}$ ways of choosing k fixed elements, the solution to the problem, denoted by $D_{n,k}$, is simply

$$D_{n,k} = \binom{n}{k} \cdot !(n-k).$$

Remark. The *Problème des Recontres* is a famous problem considered in many of the works in the field of combinatorics, including [24], [1], and [13], which was solved in 1713 by *Pierre Raymond de Montmort*. The problem itself falls into the category of partial derangement, which is illustrated in the following.

Definition 3.4.1 (Partial Derangement)**.** If some, but not necessarily all, of the items are not in their original ordered positions, the configuration can be referred to as a partial derangement

3.4.1 The Matching Problem

The matching problem, sometimes referred to as Montmort's matching problem, is an old and famous problem in probability theory, which has a wide application in many disciplines.
There are many recent studies related to the method of correct matchings. For example, some experiments' purpose was to determine whether certain aspects of personality are conveyed in handwriting. The method is attempted to match character sketch of the groups of people with specimens of handwriting of those people. The number of correct matchings thus achieved is then compared with the number to be expected alone. In the case of a significant excess of the former over the latter, it would be an evidence to support that the real similarity between sketches and handwriting must be associated with the personality [9].
We introduce the topic through an example appeared in the work of *Brian Corney* and *Tom Davis* [10].

Example 3.4.2 (Seating Mix-Up and Card Matching Problem)**.** Imagine that Yankee Stadium is completely sold out, but when the ticket holders arrive, they choose seats entirely at random, what is the probability that at least one person is seated in the seat indicated by his ticket?

Solution. It turns out that there is not much difference in the answer if there are 200 seats or 5000 seats. One nice way to approach the problem is to do a few experiments which will only require a deck of playing cards.
Let us select all of the spades and all of the hearts from the deck. Now, we place all the spades in order, shuffle the hearts and deal them out underneath the spade like this:

$A\spadesuit$	$2\spadesuit$	$3\spadesuit$	$4\spadesuit$	$5\spadesuit$	$6\spadesuit$	$7\spadesuit$	$8\spadesuit$	$9\spadesuit$	$10\spadesuit$	$J\spadesuit$	$Q\spadesuit$	$K\spadesuit$
$7\heartsuit$	$5\heartsuit$	$A\heartsuit$	$9\heartsuit$	$8\heartsuit$	$Q\heartsuit$	$2\heartsuit$	$3\heartsuit$	$J\heartsuit$	$10\heartsuit$	$K\heartsuit$	$4\heartsuit$	$6\heartsuit$

Figure 2: Card Matching Example [10]

For the particular deal above, there is a match in which the person with ticket number 10 is seating at number 10 seat. We can then repeat the experiment many times to conclude the probability, and subsequently, change it to only 8 cards and repeat the same experiment.

Definition 3.4.2. A match occurs at position j if $X_j = j$. Thus, the number of matches is the random variable N_n defined mathematically by

$$N_n = \sum_{j=1}^{n} I_j \tag{9}$$

where $I_j = 1(X_j = j)$ is the indicator variable for the event of match at position j.

Let us first consider sampling with replacement, we can find that actually I_j for j = 1,. . . ,n is a sequence of n Bernoulli trials, with p = $\frac{1}{n}$, which leads that the probability distribution of N_n is a Binomial distribution.

$$P(N_n = k) = \binom{n}{k}(\frac{1}{n})^k(1-\frac{1}{n})^{n-k}, k \in \{0, 1, \ldots, n\} \tag{10}$$

As sampling with replacement is quite an easy case, let us focus on the case of real interest - sampling without replacement. To find the probability density function of N_n, we need to count the number of permutation of $\{1, 2, \cdots\cdots, n\}$ with a specific number of matches. When there is no matches, this becomes a derangement problem.

Definition 3.4.3. We denote the number of permutations of $\{1, 2, \cdots\cdots, n\}$ with exactly k matches by

$$b_n(k) = \#(N_n = k) \quad \text{for } k \in \{0, 1, \cdots, n\}, \tag{11}$$

where, by Theorem 3.2.1,

$$b_n(0) = n!\sum_{i=0}^{n}(-1)^i\frac{1}{i!}$$

is the number of derangements of $\{1, 2, \cdots\cdots, n\}$.

Theorem 3.4.1.

$$b_n(k) = \frac{n!}{k!}\sum_{i=0}^{n-k}(-1)^i\frac{1}{i!} \quad \textit{for } k \in \{0, 1, \cdots, n\} \tag{12}$$

Proof. As there are k matches, k out of n positions are fixed and the remaining $n-k$ positions are in a derangement. Hence, we can simplify this question into finding the number of derangements with $n-k$ positions, which are chosen from the total number of positions n.

$$\begin{aligned} b_n(k) &= \binom{n}{n-k} b_{n-k}(0) \\ &= \frac{n!}{k!(n-k)!}(n-k)!\sum_{i=0}^{n-k}(-1)^i\frac{1}{i!} \\ &= \frac{n!}{k!}\sum_{i=0}^{n-k}(-1)^i\frac{1}{i!} \end{aligned}$$

□

Corollary 3.4.1.1. *The probability density function of N_n when sampling without replacement is*

$$P(N_n = k) = \frac{1}{k!}\sum_{j=0}^{n-k}\frac{(-1)^j}{j!} \quad \textit{for } k \in \{0, 1, \cdots, n\} \tag{13}$$

where $P(N_n = k) \to \dfrac{e^{-1}}{k!}$ *as* $n \to \infty$

Moreover, we are also interested in the probability density function of N_n with replacement [3].

Lemma 3.4.1. *The probability density function of N_n when sampling with replacement is*

$$P(N_n = k) = \binom{n}{k}\left(\frac{1}{n}\right)^k\left(1-\frac{1}{n}\right)^{n-k} \quad \textit{for } k \in \{0, 1, \cdots, n\} \textit{ and } E(N_n) = var(N_n) \tag{14}$$

where $E(N_n) = var(N_n) = 1$

Remark. Just as in the case of sampling with replacement, the distribution of the number of matches converges to the Poisson distribution with parameter 1 as n increases. The convergence is remarkably rapid. Indeed, the distribution of the number of matches with $n = 10$ is essentially the same as the distribution of the number of matches with $n = 1000000$!

3.4.2 Example and Applications

Having discussed the fundamental material on partial derangements, we now look at two examples on the topic.

Example 3.4.3. Ten married couples are randomly paired for a game.
(a) Find the probability density function of the number of married couples who are paired together.
(b) Find the probability that at least 3 couples are paired together.

Let X be the random variable that represents the number of married couples which are paired together.

For question (a), by Lemma 3.4.1.1 the probability density function is $P(X = k) = \frac{1}{k!}\sum_{j=0}^{10-k}\frac{(-1)^j}{j!}$

For question (b),

$$\begin{aligned} P(X \geq 3) &= 1 - P(X \leq 2) \\ &= 1 - P(X = 0) - P(X = 1) - P(X = 2) \\ &= 1 - \frac{1}{0!}\sum_{j=0}^{10}\frac{(-1)^j}{j!} - \frac{1}{1!}\sum_{j=0}^{9}\frac{(-1)^j}{j!} - \frac{1}{2!}\sum_{j=0}^{8}\frac{(-1)^j}{j!} \\ &= 0.0803 \end{aligned}$$

Now consider an example stated by *Peter J. Cameron* in [8].

Example 3.4.4. On December 20, 2000, the following question was posted in the newspaper METRO. *Match each of these languages to where they are spoken:*

1. Amharic	A. Brazil
2. Farsi	B. Ethiopia
3. Portuguese	C. India
4. Telegu	D. Iran
5. Urdu	E. Pakistan

If the options for this puzzle were given in an entirely random order, how many of the five pairs of answers would line up correctly in the same row, averaged over many puzzles? What about if there were ten options in each column?

Solution. We will seek to solve this question in the case of n options in each column. (For anyone who is interested, the answer to the first question is Amharic/Ethiopia; Farsi/Iran; Portuguese/Brazil; Telegu/India; Urdu/Pakistan).
Let us first define the random variable X, denoting the number of answers lined up correctly. Now we ask, for a given random permutation of the first column, what is the probability that $X = k$? Without loss of generality, we can rearrange the first column so that every language lines up with the correct answer. We can then count the number of fixed points the random permutation has (in relation to our new rearrangement). Then, $X = k$ if and only if the permutation has exactly k fixed points. This is the *problème des recontres*, so by Example 3.4.1 we already know this can happen in $D_{n,k} = \binom{n}{k}\cdot!(n-k)$ ways. Therefore:

$$f_X(x) = \frac{D_{n,x}}{n!}$$

And therefore the average number of correctly lined up answers is:

$$E(X) = \sum_{x=0}^{n} x f_X(x) = \sum_{x=0}^{n} x\frac{D_{n,x}}{n!}$$

In particular, the answers to the specific cases posted by the newspapers (i.e $n = 5, 10$) are given by:

$$E(X|n = 5) = 1, \qquad E(X|n = 10) = 1$$

Consider, a different approach. The probability that a particular language is on the correct position is simply $\frac{1}{n}$ (as there are $(n-1)!$ permutations where a particular language is a fixed point). So let Y_i

be the random variable representing the event that the i^{th} language is fixed, with the probability mass function as follows:

$$P(Y_i = y) = \begin{cases} 1-p & \text{if } y = 0 \\ p & \text{if } y = 1 \end{cases} \tag{15}$$

where $p = \frac{1}{n}$ is the probability that the i^{th} language is fixed. Thus $Y_i \sim Bernoulli(\frac{1}{n})$. Then the expected number of fixed points is given by (using linearity of expectation):

$$E(Y_1 + Y_2 + ... + Y_n) = n \cdot \frac{1}{n} = 1$$

Remark. This solution is simpler and more complete (as it makes it clear that the answer is independent of n). However, the first approach allows us to obtain the following interesting result, for any n:

$$\sum_{x=0}^{n} x \frac{D_{n,x}}{n!} = \sum_{x=1}^{n} \frac{1}{(x-1)!} \sum_{k=0}^{n-x} \frac{(-1)^k}{k!} = 1$$

Furthermore we can deduce the following Lemma:

Lemma 3.4.2 (Expected number of fixed points)**.** *The expected number of fixed points of a random permutation is 1.*

3.5 Derangements on Chessboards

We now consider the derangement problem in a different light, where the intuition behind using a chessboard representation will be explained through an example.

Example 3.5.1. Consider a, b, c, d, where we want the number of permutations that do not have a fixed point.
We first introduce the idea of rooks as a chess terminology. A rook is a chess piece which is able to capture any other pieces on the same row or column as itself, assuming no chess piece is in between them. Therefore, the problem is equivalent to placing 4 non-capturing rooks on the chessboard as shown in Figure 3. Then we have each of the four rooks placed on a unique row and column. And each rook placement corresponds to a permutation.

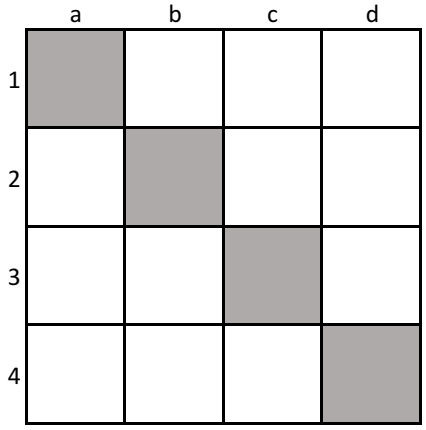


Figure 3: Chessboard

For the case that four rooks are placed on the squares $(2, a)$, $(1, b)$, $(3, c)$, $(4, d)$, it corresponds to the permutation b, a, c, d.
Letting A_i be the set of arrangements where there is a rook on a darkened square on the i^{th} column and using the Principle of Inclusion-Exclusion, we obtain

$$|\bar{A}_1\bar{A}_2\bar{A}_3\bar{A}_4| = N - S_1 + S_2 - S_3 + S_4,$$

where

$$S_1 = \sum_i |A_i| = \sum_i (1 \times 3!) = 4 \times 3!$$
$$S_2 = \sum_{i<j} |A_i A_j| = \sum_{i<j} (1 \times 2!) = \binom{4}{2} \times 2! = 6 \times 2!$$
$$S_3 = \sum_{i<j<k} |A_i A_j A_k| = \sum_{i<j<k} (1) = \binom{4}{3} = 4$$
$$S_4 = |A_1 A_2 A_3 A_4| = 1.$$

Therefore

$$|\bar{A}_1 \bar{A}_2 \bar{A}_3 \bar{A}_4| = 4! - 4! + 12 - 4 + 1 = 9$$

By Theorem 3.2.1, we can confirm that the above argument gives the correct answer, as $!4 = 4! \sum_{i=0}^{4} (-1)^i \frac{1}{i!} = 9$.

By the above example, we can say that the problem of counting derangements for a set of size k is equivalent to counting the number of different ways in which we can place k non-capturing rooks on the board, where no rook is on the diagonal.
In the following section we will generalize to the case where darkened squares are allowed to be in positions other than the diagonal, and for boards of different sizes and shapes.

4 Rook Polynomials

4.1 Introduction and Fundamental Material

Derangements can be viewed as a special case for *Rook Polynomial*, considering the board with darkened squares on the diagonal. More complicated permutation with restrictions can also be solved by applying a similar method. In this section, connection between such problem and *Rook Polynomial* will be established with an example from [1].

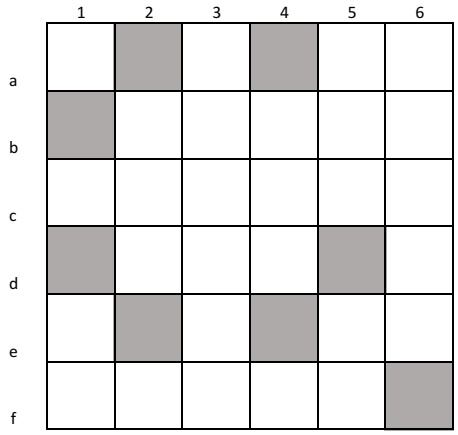


Figure 4: Chessboard C

Example 4.1.1. We shall consider the problem of finding all arrangements of a, b, c, d, e, f with the restrictions as indicated in Figure 4. That is, a may not be placed in position 2 or 4; b may not be placed in position 1; d not in 1 or 5; e not in 2 or 4; f not in 6; and there is no restriction on c. A valid arrangement can be represented by choosing six unmarked squares in the board, with no choice of squares in the same row or column.
To look at the problem in another direction, *Principle of Inclusion-Exclusion* can be applied to first determine the complementary question of choosing darkened squares that are not in the same row or column.
Let A_i be the set arrangements with a forbidden letter in position i. This is equivalent to defining A_i to be the set of arrangements with each letter being in the forbidden position. Let N be the number of all arrangements, $N = 6!$. Then, the number of all permissible arrangements can be represented as $|\bar{A}_1\bar{A}_2\bar{A}_3\bar{A}_4\bar{A}_5\bar{A}_6|$. Upon applying PIE, it becomes

$$|\bar{A}_1\bar{A}_2\bar{A}_3\bar{A}_4\bar{A}_5\bar{A}_6| = N - S_1 + S_2 + \cdots + S_6, \tag{16}$$

where $S_1 = \sum_i |A_i|$ and $S_2 = \sum_{i<j} |A_iA_j|$ etc. Notably, S_k represents the product of the number of ways to pick k darkened squares and the number of ways to permute the remaining letters, $(6-k)!$.

To proceed with solving the problem in the example, some rigorous definitions are stated from [24] and [1].

Definition 4.1.1 (Generationg Function)**.** Suppose a_r is the number of ways to select r objects in a certain procedure. Then $g(x)$ is a generating function for a_r if $g(x)$ has the polynomial expansion

$$g(x) = a_0 + a_1x + a_2x^2 + \cdots + a_rx^r + \cdots + a_nx^n \tag{17}$$

If $g(x)$ is an infinite series, it is called a **power series**[1].

Definition 4.1.2 (Rook Polynomial)**.** For a $m \times n$ chessboard C with darkened squares. Let $r_k(C)$ denote the number of ways to place k non-capturing rooks on C.

[1]Generating functions are formally discussed in [24]

Rook polynomial $R(x, C)$ of the board C, is the generating function for $r_k(C)$, and is denoted by

$$R(x, C) = r_0(C) + r_1(C)x + r_2(C)x^2 + \cdots + r_t(C)x^t, \tag{18}$$

where $t = min(m, n)$, and $r_0 = 1$.

By the above definitions, equation (16) may be rewritten as

$$|\bar{A}_1\bar{A}_2\bar{A}_3\bar{A}_4\bar{A}_5\bar{A}_6| = r_0(C) \times 6! - r_1(C) \times 5! + r_2(C) \times 4! + \cdots + r_6(C). \tag{19}$$

Then we may use rook polynomial to compute $r_k,\ k = 1, 2, \cdots 6$. It is obvious that the board is unique up to reordering of rows and columns. Thus, the board may be rearranged to a nicer form through swapping rows and columns. This is shown in Figure 5 where it is a transformed board C from Figure 4.

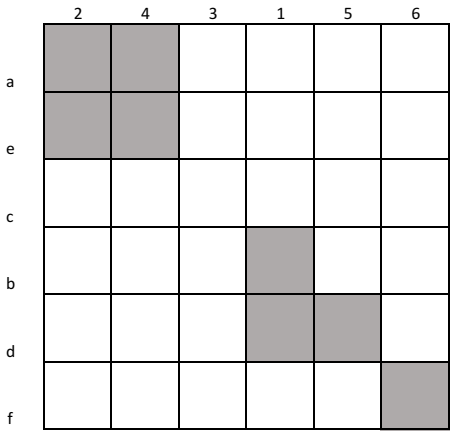


Figure 5: Rearranged Chessboard C by Rows and Columns

Let C_1 be the four darkened squares at the top-left corner, C_2 be the three darkened squares in row d and e, and C_3 the single darkened square at the bottom-right corner. Since C_1, C_2 and C_3 are disjoint by observing Figure 5, we may consider placing k non-capturing rooks on each board separately.
We can consider the case of placing two non-capturing rooks on the whole board C, then there are five disjoint cases to be considered, and it can be shown that

$$r_2(C) = r_2(C_1) + r_2(C_2) + r_1(C_1)r_1(C_2) + r_1(C_1)r_1(C_3) + r_1(C_2)r_1(C_3) \tag{20}$$

More formally, the following theorem establishes the formula of placing k non-capturing rooks on any board C which can be decomposed into two sub-boards C_1 and C_2 [1]. By induction, the result can then be generalised for any board which can be decomposed into n sub-boards.

Theorem 4.1.1 (Multiplication of Disjoint Rook's Polynomials)**.** *Let C be a board, which decomposes into two disjoint sub-boards C_1 and C_2 then:*

$$R(x, C) = R(x, C_1)\, R(x, C_2)\,, \tag{21}$$

Proof. First we consider $r_k(C)$, the number of ways of placing k non-capturing rooks in board C. In particular, we have k non-capturing rooks on board C if and only if we have $k - m$ non-capturing rooks on board C_1 and m non-capturing rooks on board C_2 for some $m \in \{0, 1, \cdots, k\}$ as the sub-boards, C_1 and C_2 are disjoint. And so, summing across all possible values for m gives:

$$r_k(C) = \sum_{m=0}^{k} r_{k-m}(C_1)r_m(C_2)$$

As for each m, there are $r_{m-k}(C_1)$ ways of arranging the rooks on C_1 and $r_m(C_2)$ ways of arranging the rooks on C_2.
The rooks polynomials of each board are

$$R(x, C_1) = r_0(C_1) + r_1(C_1)x + r_2(C_1)x^2 + \cdots$$
$$R(x, C_2) = r_0(C_2) + r_1(C_2)x + r_2(C_2)x^2 + \cdots$$

We now consider the coefficient of x^q in $R(x, C_1)R(x, C_2)$ which is

$$r_0(C_1)r_q(C_2) + r_1(C_1)r_{q-1}(C_2) + \cdots + r_q(C_1)r_0(C_2) = r_q(C).$$

Hence, by the definition of $R(x, C)$ we are done. □

Continuing from Example 4.1.1, it is fairly easy to compute $R(x, C_1)$, $R(x, C_2)$, $R(x, C_3)$ by hand, and we obtain:

$$R(x, C_1) = 1 + 4x + 2x^2,\ R(x, C_2) = 1 + 3x + x^2 \text{ and } R(x, C_3) = 1 + x,$$

thus, by the above theorem,

$$\begin{aligned} R(x, C) &= R(x, C_1)R(x, C_2)R(x, C_3) \\ &= (1 + 4x + 2x^2)(1 + 3x + x^2)(1 + x) \\ &= 1 + 8x + 22x^2 + 25x^3 + 12x^4 + 2x^5. \end{aligned}$$

Finally, by substituting r_k back into equation (19), the number of ways arranging a, b, c, d, e, f with restrictions defined by Figure 4 can be computed.
Formalising the procedure to compute the above example, we state and prove the following theorem [1].

Theorem 4.1.2 (Number of Arrangements)**.** *The number of ways to arrange n distinct objects when there are restricted positions is equal to:*

$$\sum_{k=0}^{n} (-1)^k r_k(C)(n - k)!$$

Proof. Let A_i be the set of arrangements where there is a prohibited object in the i^{th} position. And so, by the Principle of Inclusion-Exclusion we have that the number of ways to arrange the objects is:

$$|\bar{A}_1\bar{A}_2 \cdots \bar{A}_n| = N - S_1 + S_2 + \cdots + (-1)^n S_n$$

Where $S_1 = \sum_i |A_i|$ and $S_2 = \sum_{i<j} |A_iA_j|$ and so on. Let π_k be the set of all possible combinations of k elements (without repetition, and up to reordering) of the set $\{1, 2, \cdots, n\}$, and let $\alpha(j_1, \cdots, j_k) = |A_{j_1} \cdots A_{j_k}|$. Hence, $S_k = \sum_{x \in \pi_k} \alpha(x)$.
For $x = (j_1, \cdots, j_k) \in \pi_k$, we have:

$$\alpha(x) = |A_{j_1} \cdots A_{j_k}| = \beta(x)(n - k)!$$

where $\beta(x)$ is the number of different arrangements of k rooks such that there are prohibited objects on $j_1, \cdots, j_k$ positions. Therefore,

$$S_k = (n - k)! \sum_{x \in \pi_k} \beta(x)$$

However, $\sum_{x \in \pi_k} \beta(x)$ is simply the number of all possible arrangements of k rooks such that they all lie on a restricted position, so by definition it must equal $r_k(C)$. Therefore $S_k = r_k(C)(n - k)!$, and we are done. □

4.2 Generating Rook Polynomial

In this section, methods for generating rook polynomial will be introduced. The direct method of computing using *Recurrent Relation* is discussed by *John Riordan* [24] whilst a more generalised approach is developed by *Abigail G. Mitchell* [21]. Then we may look at the generating process from another perspective, by considering the bipartite graph representation of the board, and to decompose by choosing one cell at a time.

4.2.1 Recurrent Relation

We shall first formalise the definitions of board and cell for the remaining sections.

Definition 4.2.1. A board C is an $m \times n$ matrix with binary entries. Thus, $C = (c_{i,j})$. A placement of k rooks on C corresponds to a choice of k independent 1's in C. A cell is defined as the entry $c_{i,j} = 1$ in the matrix form of the board [21].
For simplicity, we only consider the cells of a board. Therefore, in the context, a board can represent the cells only.

Definition 4.2.2 (Subboard)**.** A subboard S of board C is a set of cells in C inheriting column and row relationships of C. In a matrix notation, S is represented by $(s_{i,j})$ over F_2, with injective mapping ϕ_1 and ϕ_2 from the row and column entries of S to those of C. [21]

For any board, an expression with respect to any cell can be found. Consider the following cases:
If the given cell is included, no other rook may be in a cell in the same row or column. The resulting board is a subboard generated by removing the row and column of the cell, which may be indicated by C_i, with i for 'inclusion of a given cell'.
On the contrary, if a cell is not included, undarkening the cell to obtain C_e, with e for'exclusion of a given cell'. Therefore,

$$r_k = r_{k-1}\left(C_i\right) + r_k\left(C_e\right). \tag{22}$$

which can be applied recurrently to any cell in the subboard. More formally, we shall introduce the following theorem introduced first by *John Riordan* in [24].

Theorem 4.2.1 (Cell Decomposition Theorem)**.** *Let C be a board, and $c_{i,j}$ be a cell of C. Let C_e denote the subboard obtained from C by deleting $c_{i,j}$. Let C_i denote the subboard obtained from C by deleting row i and row j. Then*

$$R(x, C) = R\left(x, C_e\right) + xR\left(x, C_i\right), \tag{23}$$

Proof.

$$\begin{aligned} R(x,C) &= \sum_k r_k(C)x^k = \sum_k r_k(C_e)x^k + \sum_k r_{k-1}(C_i)x^k \\ &= \sum_k r_k(C_e)x^k + x\sum_{k-1} r_{k-1}(C_i)x^{k-1} \\ &= R\left(x, C_e\right) + xR\left(x, C_i\right) \end{aligned}$$

□

With the result developed before, rook polynomial can be obtained for any board. If a block of cells is non-decomposable, apply the recurrent relation repeatedly until the rook polynomial can be written out with product rule of two distinct subboards.
Alternatively, a given board may be expanded with respect to any number of cells, using Equation (17). Let the darkened cells be numbered as $1, 2, 3, \cdots$, and e_k for which the k^{th} cell is excluded, and i_k for which the k^{th} cell is included,

$$R(x, C) = R[(e_1 + xi_1)(e_2 + xi_2)\cdots], \tag{24}$$

where right hand side of the above equation is an abbreviation of the following:

$$\begin{aligned} R(e_1e_2\cdots) &+ x[R(i_1e_2e_3\cdots) + R(e_1i_2e_3\cdots) + \cdots] \\ &+ x^2[R(i_1i_2e_3\cdots) + R(i_1e_2i_3\cdots) + \cdots] \\ &+ \cdots \end{aligned}$$

The coefficient of order k is equivalent to r_k. By observing row and column relations between cells chosen to decompose the board, terms may be deleted. A more systematic method will be presented later.

4.2.2 Block Decomposition Algorithm

In this section, method of choosing multiple cells for board-decomposition will be introduced. Such method will significantly reduce the number of steps taken to compute rook polynomial.

The *Block Decomposition Algorithm* was first discussed by *Abigail G. Mitchell*. The following definitions as well as the theorem are introduced in her paper [21].

Definition 4.2.3 (Covering)**.** A subboard S of C is said to cover row i of C if i is contained in the image of ϕ_1; Similarly, for S to cover column j.

In order to simplify the computation, careful choice of cells to expand on is of great importance. Symmetric property of the complement for subboard of choice is desirable. Such subboard is defined as follows.

Definition 4.2.4 (Block)**.** S is a block of C if it is a subboard of C with following constraints:

1. For any row i, i' covered by S, and any column j, j' not covered by S, $b_{i,j} = b_{i',j}$;

2. For any column j, j' covered by S, and any row i, i' not covered by S, $b_{i,j} = b_{i,j'}$

Definition 4.2.5 (Inclusion Board)**.** Let C be a board with dimension $m \times n$, and S be a subboard of C defined as above, covering s rows and t columns. For $0 \leq k \leq min(s,t)$, Let $C_{S,k}$ denote the board obtained from C by deleting

1. k of the rows covered by S

2. k of the columns covered by S

3. all the cells of S

Such board $C_{S,k}$ is called the k^{th} inclusion board with respect to S.

Notably, the inclusion board is well defined, as the rows and columns involved are identical, and all cells on S is deleted.

The following theorem is the formalisation of what is proposed at the end of the previous section.

Theorem 4.2.2 (Block Decomposition Theorem)**.** *Let C be a board with dimension $m \times n$, and S be a subboard of C defined as above, covering s rows and t columns. Let $r_k(S)$ be the coefficient of x^k in the rook polynomial of S. Let $C_{S,k}$ be the k^{th} inclusion board of C with respect to S, for $0 \leq k \leq min(s,t)$*

$$R(C) = \sum_{j=0}^{min(s,t)} r_j(S) x^j R(C_{S,j}) \tag{25}$$

Proof. Consider all possible placement of k rooks, $r_k(C)$. We may segregate board C into two disjoint subboard, S and $C_{S,j}$, as defined in Definition 4.2.4 and 4.2.5. Enumerating the value of j, we obtain all possible ways of partitioning board C, which can be viewed as $k+1$ classes with $j = 0, 1, \cdots, k$. Then it is clear that

$$r_k(C) = \sum_{j=0}^{k} r_j(S) r_{k-j}(C_{S,j}) \tag{26}$$

Invoking equation (18), it gives

$$\begin{aligned} R(C) &= \sum_{k=0}^{\min(m,n)} r_k(C)x^k \\ &= \sum_{k=0}^{min(m,n)} \sum_{j=0}^{k} x^k r_j(S) r_{k-j}(C_{S,j}) \\ &= \sum_{k=0}^{min(s,t)} \sum_{j=0}^{k} \left(x_j(S)x^j\right)\left(r_{k-j}(C_{S,j})x^{k-j}\right) \\ &= \sum_{j=0}^{\min(s,t)} r_j(S)x^j \sum_{k\geq j} r_{k-j}(C_{S,j})x^{k-j} \\ &= \sum_{j=0}^{min(s,t)} r_j(S)x^j R(C_{S,j}) \end{aligned}$$

And $\sum_{k\geq j} r_{k-j}(C_{S,j})x^{k-j}$ indeed satisfies the definition of rook polynomials as in Definition 4.1.2. □

Example 4.2.1. Consider the following board C in Figure 6.

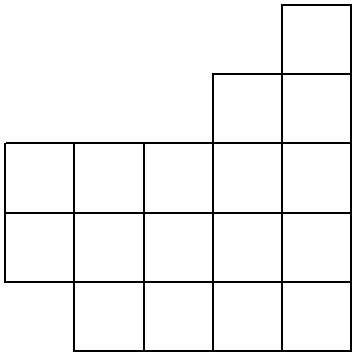


Figure 6: Board C to be decomposed

Using the method introduced in this section, let us consider choosing S. As it is easier to compute the rook polynomial for rectangular board, we may choose cells to decompose as darkened areas in Figure 7.

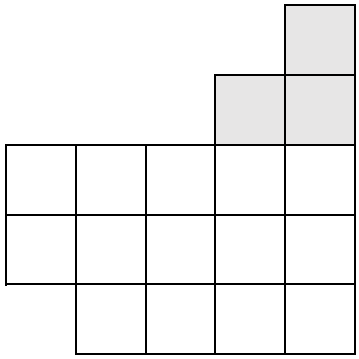


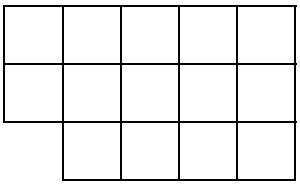
Figure 7: Board C with chosen S subboard

By Theorem 4.2.2, we can then write out the rook polynomial of the whole board as follows,

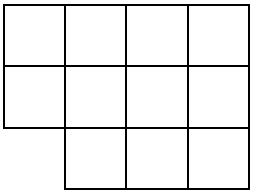
$$R(C) = r_0(S)R(C_{S,0}) + r_1(S)xR(C_{S,1}) + r_2(S)x^2R(C_{S,2}), \tag{27}$$

where subboard $C_{S,0}$, $C_{S,1}$, $C_{S,2}$ are indicated in Figure 8. Note that, all the subboards are generated according to the rule introduced in Definition 4.2.5. Then it is easy to write out the rook polynomial for C with the following relations, where $R_{m,n}$ represents a board with dimension $m \times n$.

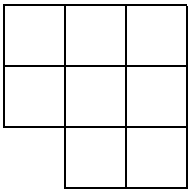
$$\begin{aligned} R(C_{S,0}) &= R_{3,5} - xR_{2,4} = 1 + 14x + 50x^2 + 48x^3 \\ R(C_{S,1}) &= R_{3,4} - xR_{2,3} = 1 + 11x + 30x^2 + 18x^3 \\ R(C_{S,2}) &= R_{3,3} - xR_{2,2} = 1 + 8x + 5x^2 \end{aligned} \tag{28}$$



(a) $C_{S,0}$



(b) $C_{S,1}$



(c) $C_{S,2}$

Figure 8: Subboards of C

The result follows directly from Equation (27) and (28).

We can see that, by applying Theorem 4.2.2, the board can be easily decomposed into more workable form, and the number of steps involved in the computation process is also reduced comparing to the *Cell Decomposition Algorithm*. However, it should be noted that it is usually difficult to find a block S for a random board, yet it is essential to the algorithm. The following example given appeared in [21] shows the complexity in spotting a workable block.

Example 4.2.2. By observing the board in Figure 9, we are able to find block S coloured as in Figure 10.

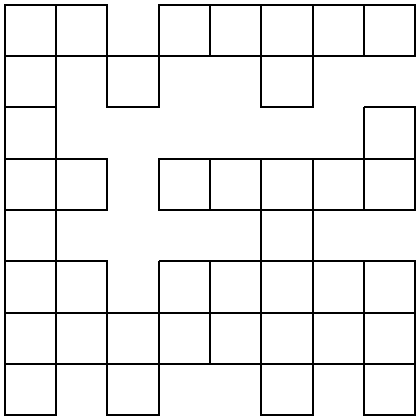


Figure 9: Complicated Example

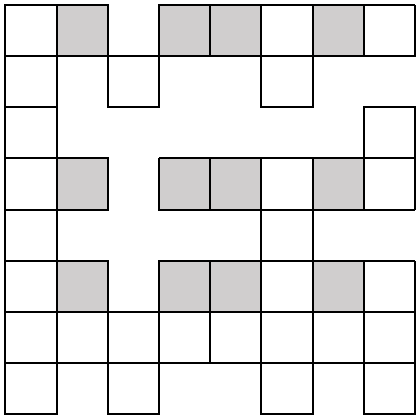


Figure 10: Complicated Example

Remark. For board C that only consists of a 1×1 cell, its rook polynomial is $1 + x$, which can be written as

$$R(C) = R(C_{S,0}) + xR(C_{S,1}),$$

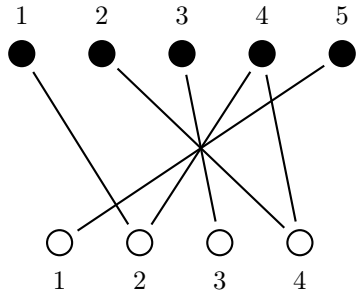
by Theorem 4.2.2, where the exclusion board and inclusion board corresponds to $C_{S,0}$, and $C_{S,1}$ respectively. Thus, in the case of 1×1 board, the *Block Decomposition Algorithm* agrees with *Cell Decomposition Algorithm*.
However, in both cases, the algorithm is greatly affected by the choice of cell or block by which to decompose.
In the following section, we will be looking at a different way of decomposing the board, by considering the bipartite graph representation of the board.

4.2.3 Improved Cell Decomposition Algorithm

In view of efficiency, it is tempting to construct an algorithm that divide the board into the form of $1 \times \star$ or $\star \times 1$ with minimum number of cells chosen by which to decompose.
Intuitively, we may consider choosing cell each at a time so that the row and column of such a cell contains the most number of cells.
To look at the problem more straightforwardly, we may convert the board into a bipartite graph on vertex sets $\{1, 2, \cdots, m\}$ and $\{1, 2, \cdots, n\}$, where the board has m rows and n columns. Then, the cell can be represented as an ordered pair (k, l), which is the edge between the vertex k and l.

Definition 4.2.6 (Bipartite Graph)**.** An unweighted, undirected graph with no graph loops or multiple edges, $G = (V, E)$, is called bipartite if its vertex set can be partitioned into two disjoint subsets $V =$

$V_1 \cup V_2$, such that every edge has the form $e = (a, b)$, where $a \in V_1$ and $b \in V_2$.[2]
e.g.



For the rest of this section, we may use an example to illustrate the computation process with the improved algorithm.

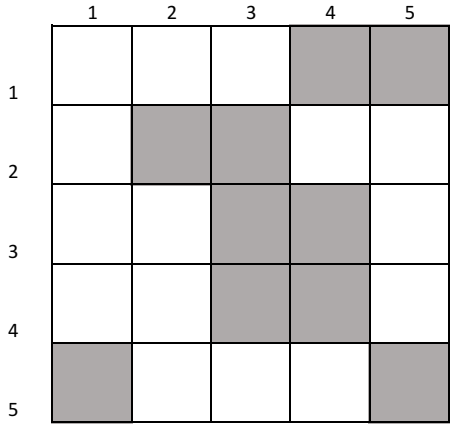


Figure 11: Chessboard D

Example 4.2.3. Consider the board in Figure 11, it can be converted into the bipartite graph as in Figure 12.

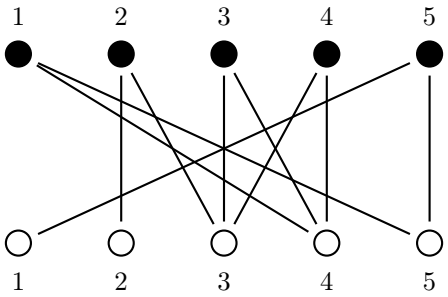


Figure 12: Bipartite Graph of D

Definition 4.2.7 (Row/Column List)**.** For a board D of dimension $m \times n$, row list is the ordered set of row indices, denoted by

$$I_m(D) = \{1_i, 2_i, \cdots, m_i\},$$

where k_i is the row index, which has a list consisting column indices of cells in row k, denoted by

$$\widetilde{k_i} = [\text{column indices of cells in } k^{th} \text{ row of the board}]$$

Column list is defined similarly and is denoted by $J_n(D)$.

By the above definition, we may obtain the following notation for the example:

$$I_5(D) = \{1_i, 2_i, 3_i, 4_i, 5_i\}$$
$$J_5(D) = \{1_j, 2_j, 3_j, 4_j, 5_j\},$$

where

$$\widetilde{1_i} = [4_j, 5_j],\ \widetilde{2_i} = [2_j, 3_j],\ \widetilde{3_i} = [3_j, 4_j],\ \widetilde{4_i} = [3_j, 4_j],\ \widetilde{5_i} = [1_j, 5_j]$$
$$\widetilde{1_j} = [5_i],\ \widetilde{2_j} = [2_i],\ \widetilde{3_j} = [2_i, 3_i, 4_i],\ \widetilde{4_j} = [1_i, 3_i, 4_i],\ \widetilde{5_j} = [1_i, 5_i].$$

We may argue recursively with the following steps.

Step 1. For a board D, find k_i that has the most number of element, i.e. $\max_{I_5} |k_i|$. Organise such k_is as an ordered set I_d. Choose the first element in I_d, denoting by d_i.

Step 2. For each element l_j in d_i, find the one that has the most number of elements, i.e. $\max_{d_i} |l_j|$. Organise such l_js as an ordered set J_d. Choose the first element of J_d, denoting as d_j.

Step 3. The edge (d_i, d_j) is the cell by which to decompose. Invoking *Recurrent Formula*, we obtain both the inclusion board D_i and exclusion board D_e.

Step 4. Update row and column list of D_i:

Delete d_i in I_5 and in the list of each element of d_i; Delete d_j in J_5 and in the list of each element of d_j; Remove any element of I_5 and J_5 that has an empty list.

Step 5. Update row and column list of D_e:

Delete d_i in the list of d_j; Delete d_j in the list of d_i; Remove any element of I_5 and J_5 that has an empty list.

Step 6. Consider separately D_i and D_e as board C in Step 1, and repeat the above steps.

Step 7. Stop when row list is empty.

By the above method, we may first choose $(1_i, 4_j)$, which produces

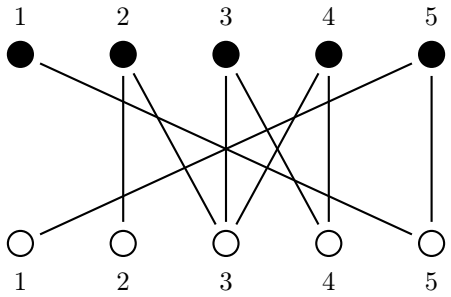


Figure 13: Exclusion Board D_e

and gives

$$R(D) = R(D_e) + xR(D_i). \tag{29}$$

Then we consider the exclusion board D_e with

$$I_5(D_e) = \{1_i, 2_i, 3_i, 4_i, 5_i\}$$
$$J_5(D_e) = \{1_j, 2_j, 3_j, 4_j, 5_j\},$$

where

$$\widetilde{1_i} = [5_j],\ \widetilde{2_i} = [2_j, 3_j],\ \widetilde{3_i} = [3_j, 4_j],\ \widetilde{4_i} = [3_j, 4_j],\ \widetilde{5_i} = [1_j, 5_j]$$
$$\widetilde{1_j} = [5_i],\ \widetilde{2_j} = [2_i],\ \widetilde{3_j} = [2_i, 3_i, 4_i],\ \widetilde{4_j} = [3_i, 4_i],\ \widetilde{5_j} = [1_i, 5_i].$$

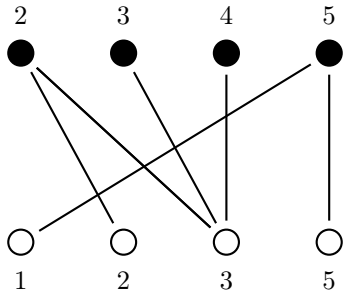


Figure 14: Inclusion Board D_i

Apply the recursive steps again, we choose $(2_i, 3_j)$ to decompose $D_1 \leftarrow D_e$ [2]. Thus we obtain

$$R(D_1) = R(D_{1e}) + xR(D_{1i}).$$

For inclusion board D_i with

$$I_5(D_e) = \{2_i, 3_i, 4_i, 5_i\}$$
$$J_5(D_e) = \{1_j, 2_j, 3_j, 5_j\},$$

where

$$\widetilde{2_i} = [2_j, 3_j],\ \widetilde{3_i} = [3_j],\ \widetilde{4_i} = [3_j],\ \widetilde{5_i} = [1_j, 5_j]$$
$$\widetilde{1_j} = [5_i],\ \widetilde{2_j} = [2_i],\ \widetilde{3_j} = [2_i, 3_i, 4_i],\ \widetilde{5_j} = [5_i],$$

we choose $(2_i, 3_j)$ to decompose $D_2 \leftarrow D_i$. And it gives

$$R(D_2) = R(D_{2e}) + xR(D_{2i}).$$

We can apply the method to find D_{1e}, D_{2e}, D_{1i} and D_{2i}, and continue the recursion until we obtain the empty board, which has rook polynomial as the constant 1. Upon completing the recursion, we may invoke backward substitution for D_k to find Equation (29).

Remark. As illustrated in the above example, the improved method of *Cell Decomposition Algorithm* introduced in this section reduces the steps of computation by refining the choice of cell. Moreover, the method can be adapted into computer algorithm, combining with the idea of a 'stacklist' [14] storing and updating boards that are to be decomposed.
Lastly, the algorithm can be realised by Python, the programme along with explanation of the code is contained in Appendix A. The result of the example generated by the programmes is also shown to be the same as that calculated by hand.

In the following sections of Section 4, we will look at special cases of rook polynomials. In Section 4.3, examples and generalisation of the matching problem is discussed. In Section 4.4, we will focus on the famous *Problèm des Ménage*.

4.3 Rectangular Boards

Let us assume for a board C with m rows and n columns for this section (n > m), where $R_{m,n}$ is its rook polynomial.

4.3.1 General Results

We are interested in obtaining the general form of rook polynomial for rectangular board C. It can be easily derived through direct reasoning as follows:
Placing k non-attacking rook on the board C requires a subboard that is of dimension $k \times k$, since we are able to rearrange the columns and rows covered by the chosen cells. It is evident that there are $k!$ ways of putting k rooks on the square board.

[2] Assign D_e to D_1

Also, a k by k board can be obtained by choosing k rows and k columns. Hence, the number of ways placing k rooks on the board C is

$$r_k = k!\binom{m}{k}\binom{n}{k} = \binom{m}{k}(n)_k \tag{30}$$

Thus by Definition 4.1.2, the rook polynomial for C can be represented as

$$R_{m,n}(x) = 1 + mnx + \binom{m}{2}(n)_2 x^2 + \cdots + (n)_m x^m \tag{31}$$

Example 4.3.1. A missile from a hostile state is travelling towards London. Trying to stop it, a MI6 spy intercepts the following code, which deactivates the missile: 7113721. Unfortunately the code has been scrambled, but not randomly as the spy knows the way the enemy thinks, and is completely sure that every number is in an incorrect position. The missile will impact in 3 minutes; and the spy can input a code every 10 seconds. Assuming the spy never repeats a code and chooses them at random, what is the probability that he can stop the missile in time?

We can solve this problem by considering the number of permutations of $7, 1, 1, 3, 7, 2, 1$ with the restricted positions as in the following chessboard:

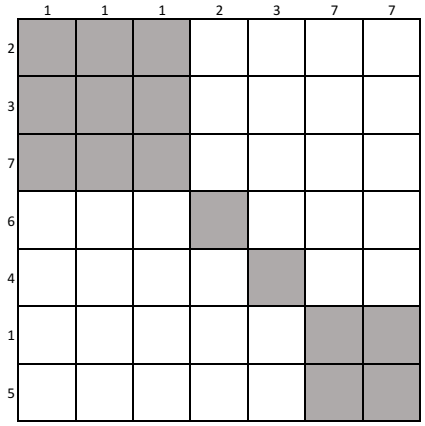


Figure 15: Board D

As we can see, it consists of four disjoint rectangular boards, hence by the general results from Rectangular Boards we can write down the Rook polynomial:

$$\begin{aligned} R(x, D) &= R_{3,3}R_{1,1}R_{1,1}R_{2,2} \\ &= (1 + 9x + 18x^2 + 6x^3)(1 + x)(1 + x)(1 + 4x + 2x^2) \\ &= 1 + 15x + 83x^2 + 221x^3 + 308x^4 + 228x^5 + 84x^6 + 12x^7 \end{aligned}$$

And therefore the number of permutations is:

$$\sum_{k=0}^{7}(-1)^k r_k(D)(7-k)! = 360$$

We now begin considering the number of codes. For each permutation of $7, 1, 1, 3, 7, 2, 1$, we can permute the three 1s in 3! ways, giving the same code. We can do the same for the 7s in 2! ways. This gives us the total number of different codes which is: $360/12 = 30$. Given his speed, the spy is able to input 17 codes in the three minutes (if he guessed the correct code at exactly three minutes, it would be too late). Therefore the probability that he stops the missile is:

$$\frac{17}{30} \approx 0.5667$$

4.3.2 Card Matching

In this section, we consider the *Card Matching Problem*, which is discussed in *Riordan*'s work in 1958 [24]. The problem involves counting enumeration of the number of matched cards in randomly ordered decks of cards.
We shall first consider the simplest situation with two decks of cards, and the number of enumeration is counted for like cards in the same position.
In this case, it shall be noted that the size of each deck is the same. As otherwise, we may choose any card to fill the missing position that is no match to cards in the other deck. Consider the example as follows:

Example 4.3.2. Suppose there are 2 decks of cards, containing three different types of cards. Then we shall mark the first type of card as a, where each deck has p_i cards; the second type of card as b, where each deck has q_i cards; and the last type of cards as c, where each deck has u_i cards, with $i = 1, 2$. Find the number of enumeration of matches.

As what is considered in this situation only concerns with the matching of the same cards in different decks, the problem can be represented neatly in the form of $m_i \times n_i$ disjoint rectangular boards, where n_i the number of i^{th} card in the first deck, m_i the second. We define A_i to be event where there is at least a match for i kind of cards in two decks', followed by what is defined above. For s kinds of cards, the rook polynomial for the whole board is

$$R(x) = R_{n_1,m_1} R_{n_2,m_2} \cdots R_{n_s,m_s}. \tag{32}$$

For one particular example, where $p_1 = p_2 = 2$, $q_1 = q_2 = 3$, $u_1 = 4$, $u_2 = 1$, the board can be represented as followed,

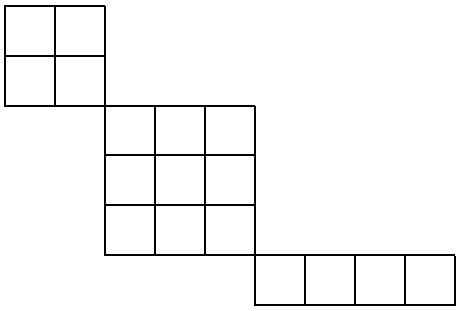


Figure 16: Board

and the rook polynomial is simply,

$$\begin{aligned} R(x) &= R_{2,2} R_{3,3} R_{4,1} \\ &= (1 + 4x + 2x^2)(1 + 9x + 18x^2 + 6x^3)(1 + 4x) \\ &= 1 + 17x + 108x^2 + 118x^3 + 444x^4 + 252x^5 + 48x^6. \end{aligned}$$

We can then obtain the answer invoking Theorem 4.1.2.
Furthermore, we may consider multiple decks of cards which is greater than two, with like cards in the same position in all decks.
An even more complicated question would be asking the matching of at least k out of n decks.

4.4 Staircase Boards

In this section, we will be introducing one of the most famous questions solved by rook polynomials, which can be represented in the form of a staircase board. More advanced discussion of the ménage problem can be found in [1] and [24].

4.4.1 Problèm des Ménage

The problem asks for the number of ways of seating n married couples in a round table with numbered chairs, women and men alternating, such that no husband is sitting next to his wife. We will solve this problem using rook polynomials.
We begin by seating the wives; it is clear there are $2 \times n!$ ways of doing this. We now have a restricted permutation problem, as the husband of the first wife cannot sit in either seat next to her. This gives us the following board to work with:

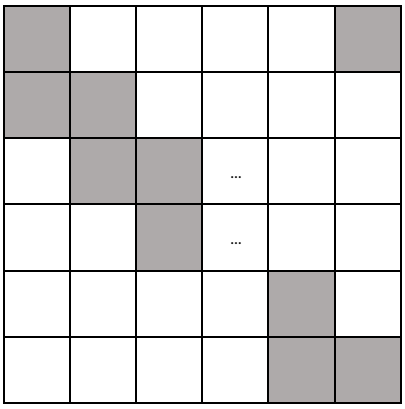


Figure 17: Ménage Chessboard: M

However we first consider the following board:

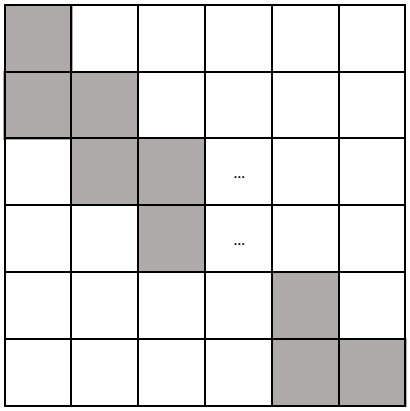


Figure 18: Ménage Chessboard: M_0

Let $L_k(x)$ be the Rook Polynomial for the first k cells of the staircase, and so, L_{2n-1} is the the polynomial of M_0. Consider the Rook's expansion of $L_k(x)$ by the first cell:

$$L_k(x) = L_{k-1}(x) + xL_{k-2}(x) \tag{33}$$

We know $L_0(x) = 1$ and $L_1(x) = 1 + x$ (the polynomial of a single cell), and so we can find $L_k(x)$:

Lemma 4.4.1. *Using $L_0(x)$, $L_1(x)$ and the recursive formula as above, we have that:*

$$L_k(x) = \sum_{j=0}^{m} \binom{k-j+1}{j} x^j \tag{34}$$

where $m = \lfloor \frac{k+1}{2} \rfloor$ [3].

[3] $\lfloor \cdot \rfloor$ the floor function

Proof. By induction on k. We can see that is is true for $k = 0$ and $k = 1$. Now assume it's true $\forall n \leq k-1$. Then:

$$\begin{aligned} L_k(x) &= L_{k-1}(x) + xL_{k-2}(x) \\ &= \sum_{j=0}^{m_1} \binom{k-j}{j} x^j + x \sum_{j=0}^{m_2} \binom{k-j-1}{j} x^j \end{aligned}$$

Case 1: k is odd. Then $m_1 = m_2 = m = (k-1)/2$. And so:

$$\begin{aligned} L_k(x) &= \sum_{j=0}^{m} \binom{k-j}{j} x^j + x \sum_{j=0}^{m} \binom{k-j-1}{j} x^j \\ &= \binom{k}{0} + \binom{k-1-m}{m} x^{m+1} + \sum_{j=1}^{m} x^j \left(\binom{k-j}{j} + \binom{k-j}{j-1} \right) \end{aligned}$$

We now note that $k - 1 - m = (2m + 1) - 1 - m = m$, and so $\binom{k-1-m}{m} = \binom{k-m}{m+1} = 1$. Furthermore:

$$\binom{k-j}{j} + \binom{k-j}{j-1} = \binom{k-j+1}{j}$$

And therefore:

$$L_k(x) = \sum_{j=0}^{m+1} \binom{k-j+1}{j} x^j$$

Case 2: k is even hence $m_1 = k/2 = m$ and $m_2 = m - 1$. And so:

$$\begin{aligned} L_k(x) &= \sum_{j=0}^{m} \binom{k-j}{j} x^j + \sum_{j=0}^{m-1} \binom{k-j-1}{j} x^{j+1} \\ &= \binom{k}{0} + \sum_{j=1}^{m} x^j \left(\binom{k-j}{j} + \binom{k-j}{j-1} \right) = \sum_{j=0}^{m} \binom{k-j+1}{j} x^j \end{aligned}$$

Hence the inductive step is done and the proof is finished. □

We now return our attention to the board M. We can expand by the top right corner giving:

$$\begin{aligned} R(x, M) &= L_{2n-1}(x) + xL_{2n-3}(x) \\ &= \sum_{j=0}^{n} \binom{2n-j}{j} x^j + x \sum_{j=0}^{n-1} \binom{2n-2-j}{j} x^j \\ &= \sum_{j=0}^{n} \frac{2n}{2n-j} \binom{2n-j}{j} x^j \end{aligned}$$

And hence, using Theorem 4.1.2, the possible number of arrangements in which the husbands can sit is given by:

$$\begin{aligned} N(M) &= \sum_{k=0}^{n} (-1)^k r_k(M)(n-k)! \\ &= \sum_{k=0}^{n} (-1)^k \frac{2n}{2n-k} \binom{2n-k}{k} (n-k)! \end{aligned}$$

Considering all possible sitting of the wives, the solution to the ménage problem is given by

$$2 \times n! \sum_{k=0}^{n} (-1)^k \frac{2n}{2n-k} \binom{2n-k}{k} (n-k)!.$$

Example 4.4.1. A doctor has 6 patients, each is taking a different medicine (call the medicines A, B, C, D, E, F). The doctor gives the drugs to a nurse, who , not knowing any better, distributes them randomly among the patients. Due to the recipients' medical conditions, it can be very dangerous if any patient takes the following drugs:

Patient 1: B, C; Patient 2: A, F; Patient 3: C, D, E; Patient 4: E; Patient 5: B, F; Patient 6: B.

Given the nurse distributed the medicines in a completely random manner, what is the probability that no patient took a drug that puts them in danger?

Notably, we can consider this as a restricted permutation problem. We look at permutations of $\{A, B, C, D, E, F\}$, not allowing A to be in the second position (as patient 2 should not take it) and so on. This gives us board D, which in turn, can be rearranged into board D_1.

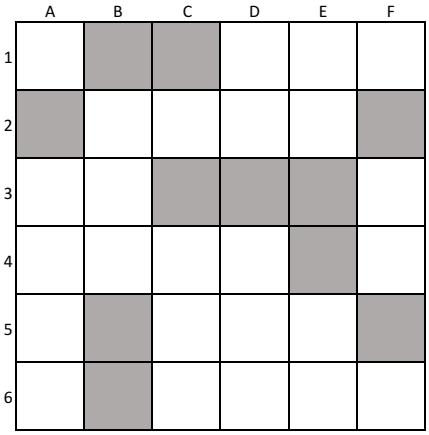


Figure 19: Board D

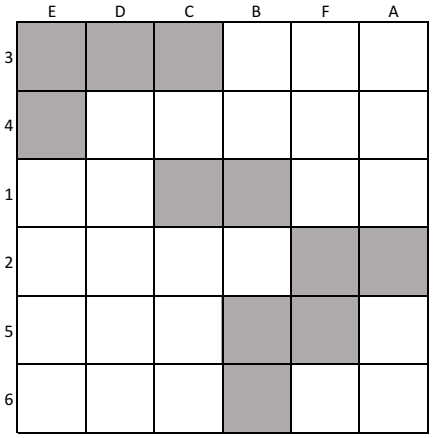


Figure 20: Board D_1

We can then expand by cell (1, B) which gives us boards D_i and D_e:

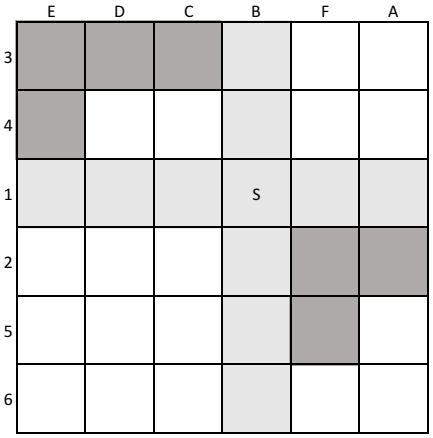


Figure 21: Board D_i

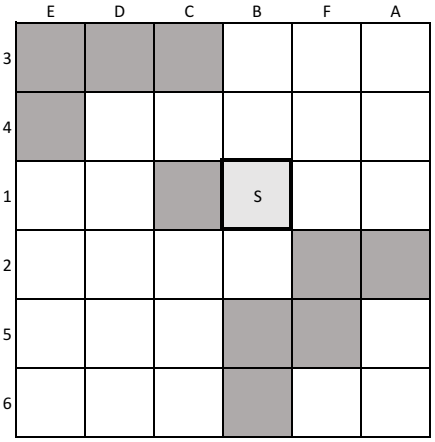


Figure 22: Board D_e

For D_i we can calculate the Rook Polynomial using the multiplication rule, giving:

$$R(x, D_i) = (1 + 4x + 2x^2)(1 + 3x + x^2) = 1 + 7x + 15x^2 + 10x^3 + 2x^4$$

Similarly, D_e consists of two disjoint boards. The polynomial for the top subboard, $R(x, D_e^{(1)})$, can be calculated by expanding on cell $(1, C)$:

$$R(x, D_e^{(1)}) = (1 + 4x + 2x^2) + x(1 + 3x + x^2) = 1 + 5x + 3x^2 + x^3$$

Then for the bottom subboard we use our work on Staircase Boards, we can easily write down the polynomial (staircase of length 5):

$$R(x, D_e^{(2)}) = 1 + 5x + 6x^2 + x^3$$

Putting both together we obtain:

$$\begin{aligned} R(x, D_e) &= (1 + 5x + 6x^2 + x^3)(1 + 5x + 3x^2 + x^3) \\ &= 1 + 10x + 34x^2 + 47x^3 + 28x^4 + 9x^5 + x^6 \end{aligned}$$

Finally, we can compute the polynomial for the whole board:

$$\begin{aligned} R(x, D) &= R(x, D_e) + xR(x, D_i) \\ &= (1 + 10x + 34x^2 + 47x^3 + 28x^4 + 9x^5 + x^6) + x(1 + 7x + 15x^2 + 10x^3 + 2x^4) \\ &= 1 + 11x + 41x^2 + 62x^3 + 38x^4 + 11x^5 + x^6 \end{aligned}$$

Using the polynomial and Theorem 4.1.2 we can calculate the number of cases in which no patient is in danger:

$$\sum_{k=0}^{6}(-1)^k r_k(D)(6-k)! = 78$$

Therefore the probability that we are looking for is:

$$\frac{78}{6!} = \frac{13}{120} \approx 0.1083$$

5 Euler's Totient Function

The Euler's Totient Function was first introduced by *Leonard Euler* in 1763 and was subsequently used in various theorems and identities such as the Euler's theorem and Menon's identity to name a few. Furthermore, the Euler's Totient plays a key role in RSA Cryptography. Henceforth, in this section we will explore the properties and applications of the Euler's Totient Function in greater details.

5.1 Introduction

Definition 5.1.1 (Euler's Totient Function)**.** We denote the Euler's Totient function, $\phi(n)$, as the number of positive integers not greater than and prime to n, that is to say the number of integers m such that $0 < m \leq n$ and $hcf(m, n) = 1$.[16]

Example 5.1.1. Let $n = 8$, then for $S = \{1, 2, \cdots, 8\}$ we have $m = 1, 3, 5, 7$ which are prime to $n = 8$ so $\phi(n) = 4$

5.2 Properties of Euler's Totient Function

Definition 5.2.1 (Multiplicative Function)**.** An arithmetic function f is multiplicative if $hcf(m, n) = 1$ implies $f(mn) = f(m)f(n)$ [17]

Theorem 5.2.1 (Multiplicativity)**.** *The Euler's Totient Function, $\phi(n)$, is a multiplicative function, i.e. if $hcf(m, n) = 1$, $\phi(mn) = \phi(m)\phi(n)$*

Proof. As published in a book by David M. Burton [7], we use the following idea for the proof of multiplicativity. Clearly, without loss of generality, for $n = 1$, since $\phi(1) = 1$, then $\phi(mn) = \phi(m) = \phi(m)\phi(1) = \phi(m)\phi(n)$. Then, for $n \neq 1$ or $m \neq 1$, we use the following table from 1 to nm to prove the above:

$$\begin{array}{ccccc}
1 & m+1 & 2m+1 & \cdots & (n-1)m+1 \\
2 & m+2 & 2m+2 & \cdots & (n-1)m+2 \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
r & m+r & 2m+r & \cdots & (n-1)m+r \\
\vdots & \vdots & \vdots & \vdots & \vdots \\
m & 2m & 3m & \cdots & nm
\end{array}$$

The table above shows an array with entries of the form $km+r$, where $0 \leq k \leq n-1$ and $0 \leq r \leq m-1$. Since $hcf(m, n) = 1$ the number of integers which are relatively prime to nm is the same as the number of integers which are relatively prime to both n and m.
Next, looking at the r^{th} row we have $hcf(km + r, m) = hcf(r, m)$ by Euclidean Algorithm and the property of congruence modulo m. Thus, integers of the form $km + r$ are relatively prime to m if and only if r are relatively prime to m. Since $0 \leq r \leq m - 1$ and by definition we have $\phi(m)$ rows of integers relatively prime to m. Subsequently, by looking at the sequence of n integers in the r^{th} row again, we can show that no two integers are congruent modulo n.
Assume for a contradiction that there exists 2 distinct integers which are congruent modulo n, then we have $km+r \equiv jm+r \pmod{n}$, where $k \neq j$ and $0 \leq k, j \leq n-1$. This then implies $km \equiv jm \pmod{n}$, and using the fact that $hcf(m, n) = 1$, the above will then simplify to $k \equiv j \pmod{n}$. However, we arrive at a contradiction as $k \neq j$ and $0 \leq k, j \leq n - 1$ so $k \not\equiv j \pmod{n}$. We shall first define $A = \{km+r : 0 \leq k \leq n-1\}$ and $B = \{0, 1, \cdots, n-1\}$. Then, we can create a bijective map, $f : A \to B$, defined as $a_i \equiv b_i \pmod{n}$ Thus, in each row there are the same number of integers which are relatively prime to n, which is namely $\phi(n)$ integers. Putting together the two implications, the total number of entries in the array which are relatively prime to both n and m, and consequently nm is the product $\phi(n)\phi(m)$. □

Theorem 5.2.2 (Prime Power Argument)**.** *If p is prime and $\alpha \geq 1$, then $\phi(p) = p - 1$ and $\phi(p^\alpha) = p^\alpha - p^{\alpha-1}$.*

Proof. By the definition of a prime number, p is prime if and only if its only divisors are 1 and itself. Hence, in the set of integers $\{1, 2, \cdots, p\}$, considering 1 to be relatively prime to all integers, there are a total of $p - 1$ integers relatively prime to p. Next, extending the above idea of counting as presented by *Aaron Greicius* [15], we observe that only the following set of integers $A = \{p, 2p, \cdots, (p^{\alpha-1} - 1)p, (p^{\alpha-1})p\}$ are divisors of p^α as p is prime. Consequently, since $|A| = p^{\alpha-1}$, we obtain the following result: $\phi(p^\alpha) = p^\alpha - p^{\alpha-1}$. □

5.3 General Formula

Theorem 5.3.1. *Let $n \geq 1$, then $\phi(1) = 1$ and*

$$\phi(n) = n \cdot \prod_{\substack{p|n \\ p \text{ prime}}} (1 - \frac{1}{p}) \tag{35}$$

Proof. (By properties of $\phi(n)$) For any $n > 1$, n can be expressed in its unique prime factorisation as $\mathbb{Z}$ is a unique factorisation domain with the prime numbers as its irreducible elements. Let $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$, where p_i are distinct primes and $\alpha_i \in \mathbb{N}$, then as presented by *Aaron Greicius* [15], we use the properties of Euler's Totient Function as shown in Theorem 5.2.1 and Theorem 5.2.2 to obtain the following:

$$\begin{aligned} \phi(n) &= \phi(p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}) \\ &= \phi(p_1^{\alpha_1})\phi(p_2^{\alpha_2}) \cdots \phi(p_k^{\alpha_k}) \\ &= (p_1^{\alpha_1} - p_1^{\alpha_1 - 1})(p_2^{\alpha_2} - p_2^{\alpha_2 - 1}) \cdots (p_k^{\alpha_k} - p_k^{\alpha_k - 1}) \\ &= n \prod_{i=1}^{k} (1 - \frac{1}{p_i}) \end{aligned} \tag{36}$$

□

Proof. (By Principle of Inclusion-Exclusion) Similarly, for any $n > 1$, n can be expressed in its unique prime factorisation. Let $n = \prod_{i=1}^{k} p_i^{\alpha_i}$, where p_i are distinct primes and $\alpha_i \in \mathbb{N}$. As shown by *Dr Noach Dina-Picard* [11] , we let A_i be the set of integers less than n which are divisible by p_i. Hence, $|A_i| = \frac{n}{p_i}$ and for any $J \subseteq \{1, 2, \cdots, k\}$, $|A_J| = \frac{n}{\prod_{i \in J} p_i}$. Then, substituting the formula of Principle of Inclusion-Exclusion,

$$\begin{aligned} \phi(n) &= |\bar{A}_1 \bar{A}_2 \cdots \bar{A}_k| \\ &= N - S_1 + S_2 - \cdots + S_k \\ &= n - \sum_{i=1}^{k} \frac{n}{p_i} + \sum_{i \neq j} \frac{n}{p_i p_j} - \cdots + (-1)^k \frac{n}{p_1 p_2 \cdots p_k} \\ &= n \prod_{i=1}^{k} \left(1 - \frac{1}{p_i}\right) \end{aligned} \tag{37}$$

Now, we will show that equation (37) is equivalent to the required general formula by induction. Let $P(m)$ be the mathematical statement that "$n-\sum_{i=1}^{m}\frac{n}{p_i}+\sum_{i\neq j}\frac{n}{p_ip_j}-\cdots+(-1)^m\frac{n}{p_1p_2\cdots p_m}=n\prod_{i=1}^{m}(1-\frac{1}{p_i})$" for $m\in\mathbb{Z}_{\geq 1}$. Clearly, this is true for $m=1$ since $n-\frac{n}{p_1}=n(1-\frac{1}{p_1})$, so $P(1)$ holds. By mathematical induction, we assume this is true for $(k-1)\in\mathbb{Z}_{\geq 1}$ so we have the statement $P(k-1)$ such that "$n-\sum_{i=1}^{k-1}\frac{n}{p_i}+\sum_{i\neq j}\frac{n}{p_ip_j}-\cdots+(-1)^{k-1}\frac{n}{p_1p_2\cdots p_{k-1}}=n\prod_{i=1}^{k-1}(1-\frac{1}{p_i})$". Then for $P(k)$ we obtain the following results:

$$\begin{aligned}
P(k) &= n\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\
&= \Big[n\prod_{i=1}^{k-1}(1-\frac{1}{p_i})\Big]\cdot(1-\frac{1}{p_k}) \\
&= \Big[n-\sum_{i=1}^{k-1}\frac{n}{p_i}+\sum_{i\neq j}\frac{n}{p_ip_j}-\cdots+(-1)^{k-1}\frac{n}{p_1p_2\cdots p_{k-1}}\Big](1-\frac{1}{p_k}) \\
&= n\Big[1-\sum_{i=1}^{k-1}\frac{1}{p_i}+\sum_{i\neq j}\frac{1}{p_ip_j}-\cdots+(-1)^{k-1}\frac{1}{p_1p_2\cdots p_{k-1}} \\
&\qquad\qquad -\frac{1}{p_k}+\sum_{i\neq k}\frac{1}{p_ip_k}-\cdots+(-1)^k\frac{1}{p_1\cdots p_{k-1}p_k}\Big] \\
&= n-\sum_{i=1}^{k}\frac{n}{p_i}+\sum_{i\neq j}\frac{n}{p_ip_j}-\cdots+(-1)^k\frac{n}{p_1p_2\cdots p_k}
\end{aligned} \tag{38}$$

Since $P(1)$ is true and $P(k-1)\implies P(k)$, by mathematical induction, the mathematical statement "$n-\sum_{i=1}^{m}\frac{n}{p_i}+\sum_{i\neq j}\frac{n}{p_ip_j}-\cdots+(-1)^m\frac{n}{p_1p_2\cdots p_m}=n\prod_{i=1}^{m}(1-\frac{1}{p_i})$" is true for all $m\in\mathbb{Z}_{\geq 1}$ so using equation (37), we obtained the required form of $\phi(n)$. □

5.4 Examples and Extensions

Example 5.4.1. For any $n>2$, $\phi(n)$ is even.

Solution. **Case 1:** $n=2^xp_1^{\alpha_1}\cdots p_k^{\alpha_k}$ where $x\in\mathbb{Z}_{\geq 2}$, $p_i\neq 2$, is prime, and $\alpha_i\in\mathbb{Z}_{\geq 0}$. Then, by Theorem 5.3.1,

$$\begin{aligned}
\phi(n) &= n\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\
&= 2^xp_1^{\alpha_1}\cdots p_k^{\alpha_k}\cdot(1-\frac{1}{2})\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\
&= 2^{x-1}p_1^{\alpha_1}\cdots p_k^{\alpha_k}\prod_{i=1}^{k}(1-\frac{1}{p_i})
\end{aligned}$$

This is clearly even as $(x-1)\in\mathbb{Z}_{\geq 1}$.

Case 2: $n = 2^x p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ where $x = 0$ or $x = 1$, $p_i \neq 2$, is prime, and $\alpha_i \in \mathbb{Z}_{\geq 1}$. Similarly, by Theorem 5.3.1,

$$\begin{aligned}\phi(n) &= n\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\ &= p_1^{\alpha_1}\cdots p_k^{\alpha_k}\cdot\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\ &= p_1^{\alpha_1-1}\cdots p_k^{\alpha_k-1}\prod_{i=1}^{k}(p_i-1)\end{aligned}$$

This is also clearly even as $p_i - 1$ is even. Hence, by the two cases above, we have shown that $\phi(n)$ is even for any $n > 2$.

Example 5.4.2. Show that if $k > 0$, then $\phi(n) = k$ has finitely many solutions.

Solution. By Theorem 4.3.1, let $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$, so we have:

$$\begin{aligned}\phi(n) &= n\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\ k &= p_1^{\alpha_1}\cdots p_k^{\alpha_k}\prod_{i=1}^{k}(1-\frac{1}{p_i}) \\ k &= \prod_{i=1}^{k} p_i^{\alpha_i-1}(p_i-1)\end{aligned}$$

By unique factorisation of integers, k has a finite number of prime factors, and thus, finite possibilities of p_i. Moreover, for any given p_i, there exists $\alpha_i \in \mathbb{Z}_{\geq 0}$ such that $p_i^{\alpha_i-1} > k$ so $p_i^{\alpha_i-1} \nmid k$. Hence, there are finite possibilities of α_i. Together, this implies a finite number of solutions for $\phi(n) = k$.

Extension. How do we find $n \in \mathbb{Z}_{\geq 0}$ such that $\phi(n) = k$? By Theorem 5.3.1, we have the following formula:

$$\phi(n) = \prod_{i=1}^{k} p_i^{\alpha_i-1}(p_i-1) = k \tag{39}$$

By doing it systematically, since $(p_i - 1)|k$, we can consider the prime divisors of k and thus, work out the finite possible values of p_i. Then, again by considering the value of k, we deduce the possible values of α_i to be 0 or 1 if $p_i \nmid k$ and derive a higher bound for α_i. Hence, by further splitting to cases, we can deduce the possible values of such n.

Example 5.4.3 (The Necklace Problem)**.** How many different necklaces with total length of 4 beads, using solely black and white beads, can be formed?

Solution. We shall split this problem into different cases and represent it using binary sequences:
Case 1: All white beads - only 1 case (i.e. 0000)
Case 2: 1 black and 3 white beads - only 1 case (i.e. 1000)
Case 3: 2 black and 2 white bead - 2 cases (i.e. 1100, 1010)
Case 4: 3 black and 1 white beads - only 1 case (i.e. 0111)
Case 5: All black beads - only 1 case (i.e. 1111)
Hence, in total, there are 6 different necklaces as shown in Figure 23.

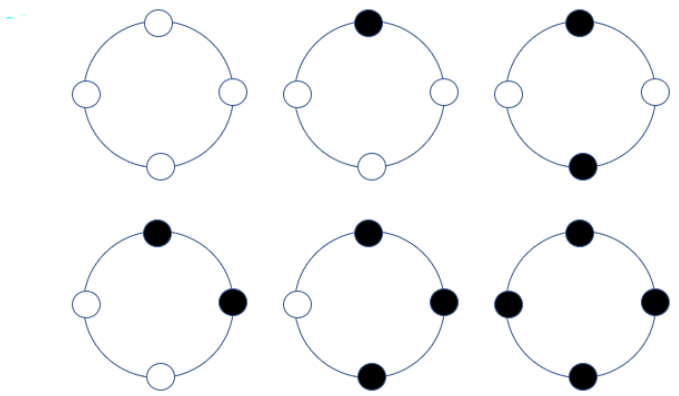


Figure 23: Possible Cases of Necklaces

Extension. For problems with greater length, n, and a greater variety of colour, m, the above counting method will not be efficient. Hence, by adapting the idea from *Muhammad Basir* and *Ansir Iqbal* [4], we shall split the problem into different cases by splitting the length of the necklace into any possible equal subsets of shorter length, l. Hence, there are $\frac{n}{l}$ number of subsets of length l. Then, there are $\phi(\frac{n}{l})$ number of elements with order $\frac{n}{l}$. Moreover, since for each bead, there are m choices of colours, then there will be m^l choices for each subset. Putting together and considering rotational symmetry, total number of different necklaces is $N_n = \dfrac{1}{n}\displaystyle\sum_{l|n} \phi(\frac{n}{l})m^l$

Example 5.4.4 (RSA Cryptography)**.** We shall explore the application of Euler's Totient Function in RSA cryptography as explained by *Clay S. Turner* [29].

Idea. The RSA cryptography involves the generation of the encryption key {e,n} and the decryption key {d,n} such that $C \equiv M^e \pmod{n}$ and $M \equiv C^d \pmod{n}$ where C is the encrypted message and M is the plain text. This is equivalent to $M^{ed} \equiv C^d \equiv M \pmod{n}$. To create such keys, we choose 2 large primes, p and q and let $n = pq$. Then, $\phi(n) = (p-1)(q-1)$ and we choose e to be a random integer co-prime to $\phi(n)$. The decryption key, d, is obtained by calculating the multiplicative inverse of $e \pmod{\phi}(n)$. This algorithm works as we have $M^{\phi(n)} \equiv 1 \pmod{n}$ by Euler's theorem. Thus, $M^{(k\phi(n)+1)} \equiv M \pmod{n}$. Then, by choosing $ed = k\phi(n) + 1 \equiv 1 \pmod{\phi}(n)$, i.e. d is the multiplicative inverse of e, we will obtain the required function of the encryption and decryption key. Lastly, by Theorem 4.2.2, $ed = k\phi(n) + 1 = k(p-1)(q-1) + 1$, hence, to obtain the decryption key, it is required to obtain the two prime factors of n. As such, the effectiveness of RSA cryptography is to select n such that it is hard to factorise into the 2 primes.

Example. From the above, we will choose our primes to be 3 and 11 to demonstrate a simple example. Primes in RSA Cryptography are generally large and unequal in length. [29] Hence, $n = 3 \cdot 11 = 33$ and $\phi(33) = \phi(3)\phi(11) = 2 \cdot 10 = 20$. Subsequently, we choose our encryption key, $e = 13$, to be co-prime to $\phi(33) = 20$, Then, $13d \equiv 1 \pmod{20} \implies d = 17$. Using the following table we have the following mapping of our RSA code:

Text	.	,	a	b	c	d	e	f	g	h	i	;	!	j	k	l	m	n	o	p	q	r	#	*	s	t	u	v	w	x	y	z	?
Plain	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Cipher	0	1	8	27	31	26	18	13	17	3	10	11	12	19	5	9	4	29	24	28	14	21	22	23	30	16	20	15	7	2	6	25	32

Table 1: Plain and Ciphertext in RSA Cryptography

As mentioned above, since the primes are relatively small, we can encode and decode using the mapping as shown in the table. However, for computational purposes, we generally use much larger primes to obtain our ciphertext using the idea shown above.

5.5 Classical Möbius Inversion

The following section aims to explore the interrelationships between the Euler's Totient and the Classical Möbius Inversion function

Lemma 5.5.1. *For each positive integer $n \geq 1$,* $\sum_{d|n} \phi(\frac{n}{d}) = \sum_{d|n} \phi(d)$

Proof. We observe that if d is a divisor of n then $\frac{n}{d}$ is also a divisor of n so we can rewrite the above as $\sum_{d|n} \phi(\frac{n}{d}) = \sum_{\frac{n}{d}|n} \phi(\frac{n}{d}) = \sum_{d|n} \phi(d)$. These are all equivalent expressions of the sum of the Euler's Totient number over all divisors of n. □

Theorem 5.5.1 (Divisor Sums)**.** *For each positive integer $n \geq 1$, we have*

$$\sum_{d|n} \phi(d) = n \tag{40}$$

Proof. (By Multiplicative Property of $\phi(n)$) We adapt the following proof from [22]. Let $F(n)$ denote the sum on the left hand side of the equation. As shown in Theorem 5.2.1, $\phi(n)$ is multiplicative, and thus, $F(n)$ is also multiplicative. Moreover, the function $g(n) = n$ is also clearly multiplicative by the multiplication operation defined on the set of all integers. Hence, to establish $F(n) = n$, this can be simplified to proving $F(p^\alpha) = p^\alpha$ where p is prime and $\alpha \in \mathbb{Z}_{\geq 0}$,

$$F(p^\alpha) = \sum_{d|p^\alpha} \phi(d) = \phi(1) + \sum_{\beta=1}^{\alpha} \phi(p^\beta) = 1 + \sum_{\beta=1}^{\alpha} (p^\beta - p^{\beta-1}) = p^\alpha$$

Lastly, for any $n \in \mathbb{Z}_{\geq 0}$, $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k}$ so from the multiplicative property,

$$\sum_{d|n} \phi(n) = F(n) = F(p_1^{\alpha_1}) \cdots F(p_k^{\alpha_k}) = p_1^{\alpha_1} \cdots p_k^{\alpha_k} = n$$

□

Proof. (By Combinatorics) As shown in a book by *David M. Burton* [7], $\Omega = \{1, 2, \cdots, n\}$ and let d be a positive divisor of n and $\pi_d = \{m | hcf(m, n) = d; 1 \leq m \leq n\}$. Then, the classes π_d are disjoint and partitions Ω. Since $hcf(m, n) = d$ if and only if $hcf(\frac{m}{d}, \frac{n}{d}) = 1$, then in each class $|\pi_d| = \phi(\frac{n}{d})$ as $|\pi_d|$ is equal to the number of positive integers not exceeding $\frac{n}{d}$ which are relatively prime to $\frac{n}{d}$. As each integer in the set Ω belongs to exactly one class π_d, we have $n = \sum_{d|n} |\pi_d| = \sum_{d|n} \phi(\frac{n}{d}) = \sum_{d|n} \phi(d)$ by Lemma 5.5.1. □

Definition 5.5.1 (Classical Möbius)**.** We shall define the Möbius function as the following:

$$\mu(n) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } p^2 | n \text{ for some prime } p \\ (-1)^r & \text{if } n = p_1 \cdots p_r \text{ for distinct primes } p_i \end{cases}$$

Lemma 5.5.2. *(Summatory Function of $\mu(n)$) As discussed by Zvezdelina Stankova-Frenkel [26], the summatory function for the Möbius function, $\mu(n)$,is defined as*

$$\sum_{d|n} \mu(d) = \begin{cases} 1 & \text{if } n = 1 \\ 0 & \text{if } n > 1 \end{cases} \tag{41}$$

Proof. Consider the case of $n = 1$, then $\sum_{d|1} \mu(d) = \mu(1) = 1$.

Next, consider the case of $n > 1$, where $n = p_1^{\alpha_1} \cdots p_k^{\alpha_k} > 1$. Hence,

$$\sum_{d|n} \mu(d) = \mu(1) + \sum_{k=1}^{r} \sum_{1 \leq i_1 < \cdots < i_r \leq k} \mu(p_{i_1} \cdots p_{i_r}) = \sum_{r=0}^{k} \binom{k}{r} (-1)^k = (1-1)^k = 0$$

□

Lemma 5.5.3. *Furthermore, it was further discussed by Zvezdelina Stankova-Frenkel [26] that if g is any arithmetic function and f is the sum function of g, so that $f(n) = \sum_{d|n} g(d)$ then $g(n) = \sum_{d|n} f(d)\mu(\frac{n}{d})$*

Proof.

$$\sum_{d|n} \mu(d) f(\frac{n}{d}) = \sum_{d|n} \mu(\frac{n}{d}) f(d) = \sum_{d|n} \mu(\frac{n}{d}) \sum_{d_1|d} g(d_1) = \sum_{d_1|n} g(d_1) \sum_{d_2|m} \mu(\frac{m}{d_2})$$

where $m = \frac{n}{d_1}$ and $d_2 = \frac{d}{d_1}$. By Lemma 4.5.2, the second sum is non-zero only when $m = \frac{n}{d_1} = 1 \implies d_1 = n$, so we have the required result of $g(n) = \sum_{d|n} f(d)\mu(\frac{n}{d})$. [26] □

Theorem 5.5.2. *For $n \geq 1$, we have $\phi(n) = \sum_{d|n} \mu(d)\frac{n}{d}$*

Proof. By adapting the idea from *Ben Lynn* [20] and using Theorem 5.5.1, as well as, the Möbius Inversion formula shown in Lemma 5.5.3, $g(n) = \sum_{d|n} f(d) \iff f(n) = \sum_{d|n} \mu(d) g(\frac{n}{d})$. Let $n = p_1^{\alpha_1} p_2^{\alpha_2} \cdots p_k^{\alpha_k}$. Then, $n = \sum_{d|n} \phi(d) \iff$

$$\begin{aligned} \phi(n) &= \sum_{d|n} \mu(d)\frac{n}{d} \\ &= n\Big(1 - \sum_{i=1}^{k} \frac{1}{p_i} + \sum_{i \neq j} \frac{1}{p_i p_j} - \cdots + (-1)^k \frac{1}{p_1 p_2 \cdots p_k}\Big) \\ &= n \prod_{i=1}^{k} \Big(1 - \frac{1}{p_i}\Big) \end{aligned} \tag{42}$$

□

Example 5.5.1 (The Necklace Problem)**.** Continuing from Example 5.4.3 and a similar example found in a book by *J.H. van Lint* and *R.M. Wilson* [18], we let $M(d)$ be the number of circular sequences of length d which are not periodic. Hence, $N_n = \sum_{d|n} M(d)$ and $\sum_{d|n} dM(d) = m^n$. By Möbius Inversion Formula, $nM(n) = \sum_{d|n} \mu(d) m^{\frac{n}{d}}$. Hence,

$$N_n = \sum_{d|n} M(d) = \sum_{d|n} \frac{1}{d} \sum_{l|d} \mu(\frac{d}{l}) m^l = \sum_{l|n} \frac{m^l}{l} \sum_{k|\frac{n}{l}} \frac{\mu(k)}{k} = \frac{1}{n} \sum_{l|n} \phi(\frac{n}{l}) m^l$$

Then, by Lemma 5.5.1, $N_n = \frac{1}{n} \sum_{l|n} \phi(\frac{n}{l}) m^l = \frac{1}{n} \sum_{l|n} \phi(l) m^{\frac{n}{l}}$

6 Möbius Inversion on a Partially Ordered Set

In this section we will explore a rather abstract interpretation of the PIE. Specifically we will focus on generalising the problem of counting in cases where the elements in question are ordered by some means. We will see that this problem occurs in many fields of Mathematics and show that the PIE and its surrounding applications are special cases of such a problem. For convenience, we will extend on notation and language from the previously mentioned *classical Möbius inversion* and discuss it as an analogue of the general case.

6.1 Fundamental Material

First let us generalise the notion of order.

Definition 6.1.1 (Partially Ordered Set)**.** A *partially ordered set* or *poset* is a set P with a binary relation or *partial order* $\leq$ such that the following three properties hold:

- Reflexivity: for each $x \in P$, $x \leq x$
- Antisymmetry: for all $x, y \in P$ such that $x \leq y$ and $y \leq x$ we have $x = y$
- Transitivity: for all $x, y, z \in P$ such that $x \leq y$, $y \leq z$ we have $x \leq z$

[18]
For convenience, in the context of partially ordered sets we will use $x < y$ to mean $x \leq y$ and $x \neq y$. Now let us generalise the Möbius inversion formula for partially ordered sets.

Definition 6.1.2 (Möbius function)**.** The *Möbius function* of a poset $(P, \leq)$ is a function $\mu : P \times P \to \mathbb{Z}$ such that for all $x, y \in P$ the following holds:

- If $x \not\leq y$ then $\mu(x, y) = 0$
- $\mu(x, x) = 1$
- If $x < y$ then $\sum_{x \leq z \leq y} \mu(x, z) = 0$

[19]
This definition may seem obscure and distant from its analogue from number theory, so I will provide a summary of its motivation. Suppose we have an arbitrary poset $(P, \leq)$ with Möbius function μ, let the function $\zeta : P \times P \to \mathbb{Z}$ be defined

$$\zeta(x, y) = \begin{cases} 1, & \text{if } x \leq y \\ 0, & \text{otherwise.} \end{cases} \tag{43}$$

So ζ indicates if one element is ordered below another according to the partial order. The three statements from the definition of the Möbius function allow for the following to hold: for all $x, y \in P$ such that $x \leq y$

$$\sum_{x \leq z \leq y} \zeta(x, z)\mu(z, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise.} \end{cases} \tag{44}$$

The right hand side is the famous Kronecker delta function on x and y. This is why the μ function is sometimes described as the *inverse* of the ζ function. The above fact is very important for proving the next theorem.

Theorem 6.1.1 (Möbius inversion formula)**.** *Let $(P, \leq)$ be a poset with Möbius function μ. Suppose $f : P \to \mathbb{C}$, $g : P \to \mathbb{C}$ are functions such that for each $y \in P$, $g(y) = \sum_{x \leq y} f(x)$ then*

$$f(y) = \sum_{x \leq y} \mu(x, y)\, g(x)$$

[6]
This formula allows us to *invert* the sum on f.

Proof (from [6, p. 190]). Suppose $(P, \leq)$, f and g are as supposed in the statement of the theorem and fix $y \in P$. Also suppose ζ is as stated in equation (43). Then

$$\begin{aligned}\sum_{x\leq y} \mu(x,y)\, g(x) &= \sum_{x\leq y} \mu(x,y) \sum_{z\leq x} f(z) \\ &= \sum_{x\leq y} \mu(x,y) \sum_{z\in P} \zeta(z,x)\, f(z) \\ &= \sum_{z\in P} f(z) \sum_{z\leq x\leq y} \zeta(z,x)\, \mu(x,y) \\ &= f(y)\end{aligned}$$

by equation (44). □

Given the simplicity of its proof one might wonder how such an important idea arises. However, the proof relies on the existence of the Möbius function which in many settings is difficult to obtain and often involves the application of induction to find an easily calculable form. Next we will explore some examples of posets with Möbius functions that we can easily evaluate.

6.2 Examples

Example 6.2.1 (Positive Integers ordered with their usual order)**.** Consider the set $\mathbb{Z}^+$ ordered by the usual notion of $\leq$, let us find the corresponding Möbius function μ.
Suppose $x, y \in \mathbb{Z}$. In the case that $x \geq y$ the value of $\mu(x, y)$ can be determined directly from Definition 6.1.2. From the first two parts of the definition we can deduce that if $x > y$ then $\mu(x, y) = 0$ and if $x = y$ then $\mu(x, y) = 1$.
Now let us consider when $x < y$, first starting with $y = x + 1$. Applying the last part of definition 6.1.2 we get

$$\begin{aligned}\mu(x,x) + \mu(x,x+1) &= 0 \\ \implies \mu(x,x+1) &= -1\end{aligned}$$

Hence for $y \geq x + 2$ we have

$$\begin{aligned}\sum_{x+2\leq z\leq y} \mu(x,z) &= -\mu(x,x) - \mu(x,x+1) + \sum_{x\leq z\leq y} \mu(x,z) \\ &= 0\end{aligned}$$

With this equation we can apply induction to show that for $y \geq x + 2$, $\mu(x, y) = 0$. Hence we have examined all cases:

$$\mu(x,y) = \begin{cases} 1, & \text{if } y = x \\ -1, & \text{if } y = x+1 \\ 0, & \text{otherwise.}\end{cases}$$

And the Möbius inversion formula gives us the following:

$$\begin{aligned}f(y) &= \sum_{x\leq y} \mu(x,y) \sum_{z\leq x} f(z) \\ &= \sum_{z\leq y} f(z) - \sum_{z\leq y-1} f(z)\end{aligned} \tag{45}$$

Though this example results in an uninteresting inversion it serves an important purpose; the poset given above is the most fundamental and well understood ordered set used in Mathematics. For that reason this example can be used to obtain an intuition for the more abstract posets explored.

Let us next discuss an example stated in Erin

Example 6.2.2 (Subsets ordered by inclusion)**.** [27] Consider an arbitrary finite set S and let $\mathcal{P}(S) = \{A : A \subseteq S\}$ be its power set and suppose we have the relation of inclusion $\subseteq$. Using basic set theoretic facts we can show that this relation fulfils all requirements in Definition 6.1.1 and hence with P it forms a poset. Suppose μ is its Möbius function of which we will be evaluating for an arbitrary $A, B \in P$. As in Example 6.2.1, the only case in which the value of $\mu(A, B)$ is not derived directly from Definition 6.1.2 is the case $A \subsetneq B$. I claim that in this case $\mu(A, B) = (-1)^{|B|-|A|}$ and I will prove it by strong induction on $k = |B| - |A|$.

Proof. Suppose $k = 1$, then by $A \subseteq B$ and $|B| - |A| = 1$ we gather that there is no subset $C \in P$ such that $A \subsetneq C \subsetneq B$. By $\mu(A, A) + \mu(A, B) = 0$ we have

$$\mu(A, B) = -1$$

and hence the equation holds for the base case.
Now suppose it holds for all k up to $k = n - 1$ and consider the case in which $k = n$. Since $|B \setminus A| = n$ there are $\binom{n}{x}$ many subsets of $B \setminus A$ of size x. Also, for each $C \in P$ such that $A \subseteq C \subseteq B$ we can write it as a union of disjoint sets $C = A \cup K$ where $K \subseteq B \setminus A$. So there are $\binom{n}{x}$ many sets C such that $A \subseteq C \subseteq B$ and $|C| - |A| = x$ and by the induction assumption

$$\begin{aligned}
\mu(A, B) &= -\sum_{A \subseteq C \subsetneq B} \mu(A, C) \\
&= -\sum_{A \subseteq C \subsetneq B} (-1)^{|C|-|A|} \\
&= -\sum_{x=0}^{n-1} \binom{n}{x} (-1)^x \\
&= \binom{n}{n} (-1)^n - \sum_{x=0}^{n} \binom{n}{x} (-1)^x \\
&= (-1)^n - (1-1)^n \\
&= (-1)^n
\end{aligned}$$

Hence the equation holds for $k = n$ and the induction argument is complete. So we have considered all cases:

$$\mu(A, B) = \begin{cases} (-1)^{|B|-|A|}, & \text{if } A \subseteq B \\ 0, & \text{otherwise.} \end{cases} \tag{46}$$

□

Since this poset poses a more abstract notion of order it seems fitting to introduce a method visualising of it. Let us consider Example 6.2.1 once again, in which we explore the familiar poset $(\mathbb{Z}, \leq)$. Upon observing Figure 24 one might describe it as a *number line*, which is a visual tool we use to understand the integers when first learning about them. However, this diagram also demonstrates a good way to represent the ordering $(\mathbb{Z}, \leq)$ since elements appear below any elements that are greater than it and is connected to them by some chain of lines. So let us generalise this for any poset with the *Hasse diagram* which is a graphical diagram of the elements of a poset constructed according to the following two rules:

1. If $x < y$ in the poset then x should appear lower than y in the diagram.
2. If $x < y$ and there exists no elements z such that $x < z < y$ then x and y are connected by a line segment

[28]
For example see Figure 25 to see the Hasse diagram for the power set of $\{1, 2\}$ with the relation $\subseteq$.

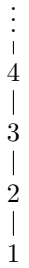


Figure 24: Hasse diagram for $(\mathbb{Z}^+, \leq)$

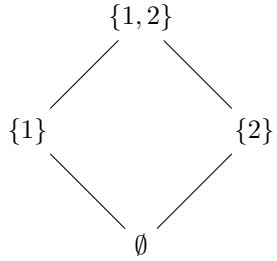


Figure 25: Hasse diagram for $(\mathcal{P}(\{1,2\}), \subseteq)$

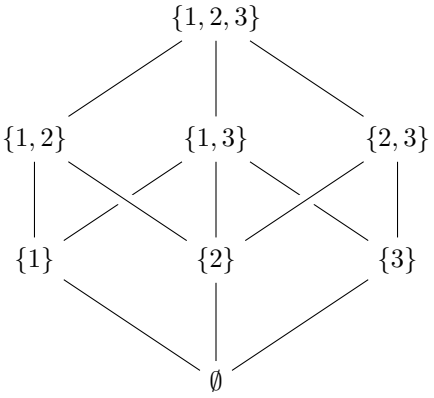


Figure 26: Hasse diagram for $\mathcal{P} = (\{1,2,3\}, \subseteq)$

Example 6.2.3 (Classical Möbius Inversion)**.** [27] Since we have adapted the notation and language of the Classical Möbius Inversion, one might expect that we can derive it as a special case of the inversion on a poset. In this example we will show exactly that.

Let n be a positive integer and $N = \{1, 2, ..., n\}$. Consider the poset $(N, |\,)$ where the partial order is given by divisibility, meaning that we say $x \leq y$ if and only if $x \mid y$ (x divides y). For example see the Hasse diagram in Figure 27 for this ordering for the divisors of 30.

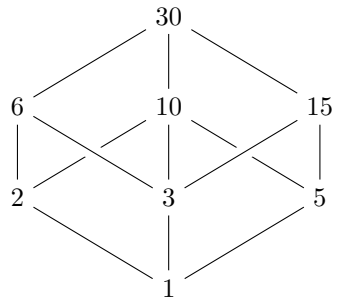


Figure 27: Hasse diagram for the divisors of 30 ordered by divisibility

For this poset the Möbius function has a useful property: $\mu(x, y) = \mu(1, \frac{y}{x})$ for all $x \leq y$. We will show this by induction on y for some fixed $x \in N$. For $x = y$ we have $\mu(x, y) = \mu(1, 1)$, hence this case holds.

Now suppose that for all $x \leq z < k$, $\mu(x,z) = \mu(1, \frac{z}{x})$ and let us consider the case $y = k$:

$$\begin{aligned}\mu(x,k) &= -\sum_{x \leq z < k} \mu(x,z) \\ &= -\sum_{x \leq z < k} \mu(1, \frac{z}{x}) \\ &= -\sum_{1 \leq \frac{z}{x} < \frac{k}{x}} \mu(1, \frac{z}{x}) \\ &= \mu(1, \frac{k}{x})\end{aligned}$$

Hence our induction argument is complete. This property greatly simplifies the task of evaluating the Möbius function as we only need to evaluate $\mu(1,m)$ for every $m \in N$. Consider the case in which m is a product of r distinct primes (i.e., m is divisible by no square other than 1). If we let S be the set of prime factors of m then we can think of every divisor of m as a product of the elements of a subset of S. So each divisor of m naturally corresponds to a subset of S, with 1 corresponding to the empty set. Continuing this analogy, the inclusion of these subsets occur if and only if the products formed by them divide and so we have an ordering that is equivalent to that of the $(\mathcal{P}(S), \subseteq)$ (see Example 6.2.2). For example, 30 is the product of three distinct primes so the ordering of its factors by divisibility is equivalent to the poset $(\mathcal{P}(\{2,3,5\}), \subseteq)$ (compare Figures 26 and 27). From this we can deduce that for $m = p_1 \cdots p_r$ where $p_1, \cdots, p_r$ are distinct prime numbers then

$$\begin{aligned}\mu(1,m) &= (-1)^{|\{p_1,\cdots,p_r\}|-|\emptyset|} \quad \text{By Equation (46)} \\ &= (-1)^r\end{aligned}$$

Now suppose m is not necessarily a product of distinct primes. Let us prove by induction that for $m = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ where $p_1, \cdots p_r$ prime and $\alpha_1, \cdots \alpha_r$ we have

$$\mu(1,m) = \begin{cases} 0, & \text{if } \alpha_i > 1 \text{ for some } i \\ (-1)^r, & \text{otherwise.} \end{cases} \tag{47}$$

The base case holds as $\mu(1,1) = 1$. Now suppose that the statement holds for all $k < m$. We know that if m can be written as the product of r distinct primes then $\mu(1,m) = (-1)^r$ so instead consider the case in which it cannot be written in this way. But m can be written as $m = p_1^{a_1} \cdots p_r^{\alpha_r}$ where $p_1, \cdots, p_r$ are prime and each α_i is a positive integer and thus

$$\begin{aligned}\mu(1,m) &= -\sum_{1 \leq k < m} \mu(1,k) \\ &= -\sum_{1 \leq k \leq p_1 \ldots p_r} \mu(1,k) \quad \text{by the induction assumption} \\ &= 0\end{aligned}$$

Hence we have proven equation (47). So we have evaluated all cases and have deduced that

$$\mu(x,y) = \begin{cases} 1, & \text{if } x = y \\ (-1)^r, & \text{if } y \mid x \text{ and } \frac{y}{x} \text{ can be written as the product of } r \text{ distinct primes} \\ 0, & \text{otherwise.} \end{cases}$$

The definition of the classical Möbius function at some n is equivalent to the above definition at $x = 1$, $y = n$. Thus the inversion formula given by this poset is equivalent to the Classical Möbius Inversion.

6.3 The PIE as a Special Case

In this section we will apply the Möbius Inversion of a poset to deduce the PIE in the following form: if we have a set S with subsets $A_1, ..., A_n$ then

$$|S \setminus \cup_{i=1}^n A_i| = \sum_{I \subseteq N} (-1)^{|I|} \left| \bigcap_{i \in I} A_I \right| \tag{48}$$

where $N = \{1, \cdots n\}$[6][27]. For tidiness we will denote $\bigcap_{i \in I} A_i$ by A_I. Naturally, the poset we should apply is the one explored in Example 6.2.2 where we ordered a power set by inclusion and obtained that the Möbius function for such posets is the same as in equation (46). So in this proof we will apply the Möbius Inversion on the poset $(\mathcal{P}(N), \subseteq)$.
To obtain a summation as neat as that seen in the PIE we must find a convenient summation to invert. First, define a function H that maps a subset of N, say $K \subseteq N$ to the set of elements which belong only to A_i for each $i \in K$. We can define this formally as

$$H(K) = A_K \setminus \bigcup_{K \subsetneq I} A_I$$

Suppose $I \subseteq N$, then from the definition of H we can show that $\{H(J) : J \supseteq I\}$ is a collection of pairwise disjoint sets of which under union form A_I (see Appendix 6.4). See Figure 28 for a visual example with three sets that demonstrates how H partitions the union in the same way that the lines of a Venn diagram partition the union. Hence we can establish the sum

$$\sum_{I \subseteq K} |H(K)| = |A_I| \tag{49}$$

Now let us define our functions on which we will perform the Möbius Inversion. For each $K \subseteq N$ let f and g be defined

$$f(K) = |H(N \setminus K)|$$
$$g(I) = \sum_{K \subseteq I} f(K)$$

So $f(K)$ is defined as the number of elements of S that belong only to the sets A_i with $i \notin K$. Applying what we know about H we can find a simple expression for g:

$$\begin{aligned} g(I) &= \sum_{K \subseteq I} f(K) \\ &= \sum_{K \subseteq I} |H(N \setminus K)| \\ &= \sum_{N \setminus J \subseteq I} |H(J)| \qquad \text{by setting } J = N \setminus K \\ &= \sum_{N \setminus I \subseteq J} |H(J)| \qquad \text{since } N \setminus J \subseteq I \Leftrightarrow N \setminus I \subseteq J \\ &= |A_{N \setminus I}| \end{aligned}$$

where the last line is achieved using equation (49). Now we apply the Möbius Inversion formula to obtain the following:

$$\begin{aligned} f(I) &= \sum_{K \subseteq I} \mu(K, I) g(K) \\ &= \sum_{K \subseteq I} (-1)^{|I| - |K|} |A_{N \setminus K}| \end{aligned}$$

Setting $I = N$ we obtain

$$\begin{aligned} f(N) &= \sum_{K \subseteq N} (-1)^{|N| - |K|} |A_{N \setminus K}| \\ &= \sum_{K \subseteq N} (-1)^{|N \setminus K|} |A_{N \setminus K}| \\ &= \sum_{J \subseteq N} (-1)^{|J|} |A_J| \end{aligned}$$

Using the our formal definition for f we also obtain

$$\begin{aligned} f(N) &= |H(\emptyset)| \\ &= |A_\emptyset \setminus \bigcup_{\emptyset \subseteq I} A_I| \\ &= |S \setminus \bigcup_{i=1}^{n} A_i| \end{aligned}$$

Hence equating these gives us the PIE formula.

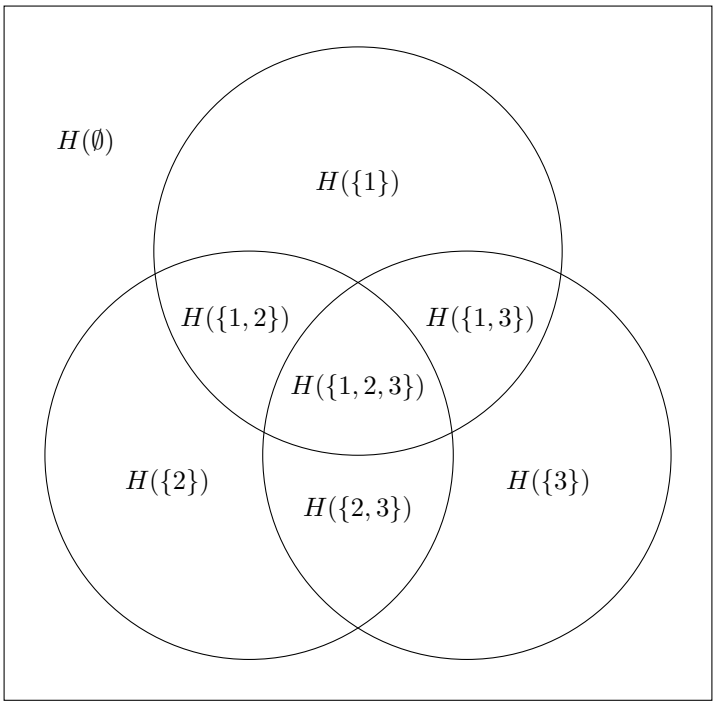


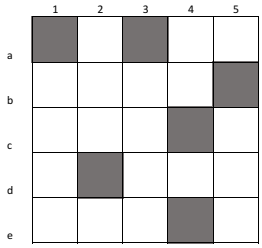
Figure 28: A diagrammatic example of how H partitions a set

In Example 6.2.1 we obtained the inversion formula

$$f(y) = \sum_{z \leq y} f(z) - \sum_{z \leq y-1} f(z)$$

for the positive integers. Note that this is an example of over-generous counting and compensating under-counting and as this proof illustrates, the PIE can also be described as exactly that. In fact, the generalised Möbius inversion can be thought of as an abstract form of such a counting procedure.
Next we will discuss an example from *Brualldi* [6]

Example 6.3.1 (Restricted Permutation Problems)**.** Let us consider a rook-placing problem on an n by n board as in Section 4. Naturally, we can represent the problem with an $n \times n$ matrix $A = (a_{ij})$ of binary entries where the value of an entry is 0 only in the corresponding positions of any darkened cells. For example see figure 29.



$$\begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Figure 29: A Rook's problem on a square board and its corresponding matrix form

Let $N = \{1, \cdots, n\}$, then we can figure out the number of permitted arrangements by instead counting the number of bijections $f : N \to N$ such that $a_{i\,f(i)} = 1$ for each $i \in N$. Thus if we let $\mathcal{B}_n$ be the set of all bijections from N to N we have that the number of permitted arrangements can be calculated by

$$\sum_{f \in \mathcal{B}_n} \prod_{i=1}^{n} a_{i\,f(i)} \tag{50}$$

Let S be a subset of N and define $\mathcal{F}_n(S)$ as the set of all functions $f : N \to S$ and define $\mathcal{G}_n(S)$ as the set of the surjective functions in $\mathcal{F}(S)$. Then we have

$$\mathcal{F}_n(S) = \bigcup_{T \subseteq S} \mathcal{G}_n(T) \tag{51}$$

Then for our inversion formula, define $F(S)$ as the number of permitted arrangements in $\mathcal{G}(S)$ and $G(S) = \sum_{T \subseteq S} F(T)$. Thus we can write these explicitly as

$$F(S) = \sum_{f \in \mathcal{G}_n(S)} \prod_{i=1}^{n} a_{i\,f(i)}$$

and

$$\begin{aligned} G(S) &= \sum_{T \subseteq S} \sum_{f \in \mathcal{G}_n(S)} \prod_{i=1}^{n} a_{i\,f(i)} \\ &= \sum_{f \in \mathcal{F}_n(S)} \prod_{i=1}^{n} a_{i\,f(i)} \quad \text{By Equation (51)} \\ &= \prod_{i=1}^{n} \left(\sum_{j \in S} a_{ij} \right) \end{aligned}$$

The last line is derived from the argument that summing over $\prod_{i=1}^{n} a_{i\,f(i)}$ for all functions $f : N \to S$ is equivalent to taking the product of the number of permitted column choices within S for each row. If we apply the Möbius Inversion formula on the poset $(\mathcal{P}(S), \subseteq)$ we obtain that

$$F(N) = \sum_{T \subseteq N} (-1)^{n-|T|} G(T)$$

Also, since every function in $\mathcal{G}(N)$ must be injective we have that $F(N)$ is exactly the formula given by Equation (50). Hence, we have obtained that the number of permitted arrangements is

$$\sum_{T \subseteq N} (-1)^{n-|T|} \prod_{i=1}^{n} \left(\sum_{j \in T} a_{ij} \right)$$

6.4 The Motivation Behind Generalising the PIE

It would be reasonable to ask why such an abstract Mathematical concept has been employed to deduce results that have already been ascertained in previous sections. To address this, we turn to a fundamental paper in the study of this inversion formula: *On the Foundations of Combinatorial Theory, I. Theory of Möbius Functions* by Gian-Carlo Rota [25]. In the introduction, Rota addresses the PIE:

> One frequently notices [...] a wide gap between the bare statement of the principle and the skill required in recognizing that it applies to a particular combinatorial problem. It has often taken the combined efforts of many a combinatorial analyst over long periods to recognize an inclusion-exclusion pattern. [...] The lack of a systematic theory is hardly matched by the consummate skill of a few individuals with a natural gift for enumeration.

The sentiment of Rota's words is demonstrated in the previous sections of this report where many interesting problems are phrased as applications of the PIE in a somewhat awkward manner. In fact many problems that had been solved by applying this framework based on over-counting and under-counting seem to have very little unifying them apart from their seemingly artificial relationship to the PIE [5]. In his paper, Rota used the Möbius inversion formula to illustrate an alternative approach for problems of this type by allowing them to be thought of as an inversion of some abstract summation.
What first appeared as a substitute for the PIE then grew into an active area of study from which many results related to graph colouring, flow in networks and lattices have been formed[On the applications]. To date, the Möbius inversion formula has become a fundamental staple in the study of combinatorics [5].

Appendix

Code for Improved Cell Decomposition Algorithm

We first develop the idea of a 'stacklist', introduced by *Daniel C. Fielder* in his paper [14].
The initial stacklist only contains the board D that we want to decompose, represented by a row and a column list defined in 4.2.7. After applying Theorem 4.2.1 to the cell we chose, using the improved algorithm, we obtain an exclusion board $D_1 \leftarrow D_e$ and an inclusion board $D_2 \leftarrow D_i$, which are appended to the bottom of the stack in order. (To be able to represent the rook polynomial, we also append a symbolic variable 'x' to the expression of inclusion board.) D is then removed from the stack, allowing D_1 to become the first level of stack. We then apply the same procedure recursively, by operating solely on the first level. If the board of first level is empty, we append it to the bottom of the stack. Operation ends with empty row and column lists for all boards in stcklist.
Finally, we have the following code that describes the algorithm, along with a solved example given by Figure 12.

```
import sympy, collections, copy
# represent the x timed in 'recurrent formula'
x = sympy.Symbol("x")

# define node and its methods for row and col lists
class node:
    def __init__(self, val, type_, list_):
        self.value = val
        self.type = type_
        self.child = list_
    def remove_child(self, del_child):
        self.child.remove(del_child)
    def add_child(self, new_child):
        self.child.append(new_child)


# example setup------------------------------------------------------------
## initialise row and col lists
rowlist = [node(i, "row", []) for i in range(1, 5 + 1)]
collist = [node(i, "col", []) for i in range(1, 5 + 1)]

c_r1 = [4, 5]
c_r2 = [2, 3]
c_r3 = [3, 4]
c_r4 = [3, 4]
c_r5 = [1, 5]

c_r = [c_r1, c_r2, c_r3, c_r4, c_r5]

## adding child to row ndoes
for i in range(len(c_r)):
    cur = c_r[i]
    for j in range(len(cur)):
        rowlist[i].add_child(cur[j])
# change row to a dictionary
rowlist = dict(zip([x for x in range(1, 5 + 1)], rowlist))
r_len = len(rowlist)

# adding child to col nodes
r_c1 = [5]
r_c2 = [2]
r_c3 = [2, 3, 4]
r_c4 = [1, 3, 4]
r_c5 = [1, 5]
```

```
r_c = [r_c1, r_c2, r_c3, r_c4, r_c5]


for i in range(len(r_c)):
    cur = r_c[i]
    for j in range(len(cur)):
        collist[i].add_child(cur[j])
# change column to a dictionary
collist = dict(zip([x for x in range(1, 5 + 1)], collist))
c_len = len(collist)

#--------------------------------------------------------------------------


## initialise stack list
## check that the row node is deleted for those with empty child list
stack = []
for i in range(1, r_len + 1):
    if rowlist[i].child == []:
        rowlist[i] = []

for i in range(1, c_len + 1):
    if collist[i].child == []:
        collist[i] = []

stack.append([rowlist, collist])

res_r = 1


# Find the deleting cell for given row and col list of nodes
def delete(row, col):
    ## find the list of rows who has the max children
    size_r = []
    for i in range(1, r_len + 1):
        r = row[i]
        if type(r) == node:
            size_r.append(len(r.child))
        else:
            size_r.append(0)

    ## the row of the cell by which to decompose
    val_r = size_r.index(max(size_r)) + 1
    del_r = row[val_r]

    ## find the list of cols who has the max children
    size_c = []
    set_c = del_r.child
    for i in range(1, c_len + 1):
        c = col[i]
        if type(c) == node:
            if c.value in set_c:
                size_c.append(len(c.child))
            else:
                size_c.append(0)
        else:
            size_c.append(0)

    ## the col of the cell by which to decompose
```

```
    val_c = size_c.index(max(size_c)) + 1
    del_c = col[val_c]

    return (del_r, del_c)

# Build basic block, 'Recurrent Formula', for recusion

# find the row and col list of inclusion board
def recurrent_i(board):
    row_i, col_i = (board[0], board[1])
    ## find cell of the board by which to decompose
    del_r, del_c = delete(row_i, col_i)
    ## update row list
    row_i[del_r.value] = []
    ## update element of col list's child list
    ## delete del_r in all child list of col nodes
    ## if child list empty, replace with empty list in col list
    for c in del_r.child:
        col_i[c].remove_child(del_r.value)
        if col_i[c].child == []:
            col_i[c] = []

    ## Similarly, update col list

    ## update the del_c
    del_c = col_i[del_c.value]
    if type(del_c) == node:
        col_i[del_c.value] = []
        ## update element of row list's child list
        for r in del_c.child:
            row_i[r].remove_child(del_c.value)
            if row_i[r].child == []:
                row_i[r] = []

# find the row and col list of exclusion board
def recurrent_e(board):
    row_e, col_e = (board[0], board[1])
    ## find cell of the board by which to decompose
    del_r, del_c = delete(row_e, col_e)

    ## update elements of col and row list's child list

    col_e[del_c.value].remove_child(del_r.value)
    if col_e[del_c.value].child == []:
        col_e[del_c.value] = []

    row_e[del_r.value].remove_child(del_c.value)
    if row_e[del_r.value].child == []:
        row_e[del_r.value] = []



# recursion step
# always process the first level of the stack
def block_stack(stack):
    ## get seperate copies of row and col lists for inclu and exclu board
    board = stack[0]
    board_e = copy.deepcopy(board)
    board_i = copy.deepcopy(board)

    ## generate a new exclusion board
```

```
    recurrent_e(board_e)
    list_e = copy.deepcopy(board)
    ## update the row and col in the first level of stack
    list_e[0] = board_e[0]; list_e[1] = board_e[1]
    stack.append(list_e)

    ## similar as above
    recurrent_i(board_i)
    list_i = copy.deepcopy(board)
    list_i[0] = board_i[0]; list_i[1] = board_i[1]
    ## add another x for inclusion board
    list_i += [x]
    stack.append(list_i)

    ## remove the 1st level stack
    stack.remove(stack[0])

    ## compute the length of children's list for each row node
    board_ = stack[0]
    row_ = board_[0]

    res_r = 0
    for i in range(1, r_len + 1):
        r = row_[i]
        if type(r) == node:
            res_r += 1
        else:
            res_r

    return(stack, res_r)


# recursion to find coeffs

## recursion
def build_stack(res_r, stack):
    ## run block_stack while row list not empty (res_r not 0)
    while res_r > 0:
        res_r = block_stack(stack)[1]

    i = 0
    ## if len_r is 0
    for level in stack:
        ## update the number of loopings
        i += 1

        row_ = level[0]

        list_node = []
        for k in range(1, r_len + 1):
            r = row_[k]
            list_node.append(type(r) == node)

        ## check if all r in row has only empty list
        if any(list_node) == True:
            ### update stack
            remove_level = copy.deepcopy(stack[0:i - 1])
            stack[0:i - 1] = []
            stack += remove_level
            break
        else:
```

```
            if i == len(stack):
                return stack

    ## update len_r
    for k in range(1, r_len + 1):
        r = row_[k]
        if type(r) == node:
            res_r += 1
        else:
            res_r
    ## restart the recursion for the new first level
    build_stack(res_r, stack)


## count the freq for each order of x
def count_ord(stack):
    ### remove the row and col in each level of the stacks
    for level in stack:
        level.remove(level[0])
        level.remove(level[0])

    ### store the coeff in a dictionary to access
    len_ = [len(i) for i in stack]
    counter = collections.Counter(len_)
    coeff = dict(counter)

    return coeff
```

We then get the following output using command line,

```
build_stack(res_r, stack)
count_ord(stack)

> {0: 1, 1: 10, 2: 33, 3: 42, 4: 18, 5: 2}
```

which is a dictionary containing the coefficient of rook polynomial of board D represented by Figure 12.

Partitioning a set

Suppose we have a set S with subsets $A_1, ..., A_n$. Let $N = \{1, ..., n\}$. and $H : N \to \mathcal{P}(S)$ be a function such that for each $K \subseteq N$

$$H(K) = A_K \setminus \bigcup_{K \subseteq I} A_I$$

Then for each $J \subseteq N$, we have that

$$\{H(K) : K \supseteq J\}$$

is a partition of A_J.

Proof. Let J be some subset of N. Suppose $x \in A_J$, then we can choose the largest set $K' \subseteq N$ such that $x \in A_{K'}$. Hence $x \notin \bigcup_{K' \subsetneq I} A_I$ and so $x \in H(K')$. Also, for each $K \supseteq J$, $A_K \subseteq A_J$ and so $H(K) \subseteq H(J)$. Thus, we have deduced that

$$A_J = \bigcup_{K \supseteq J} H(K)$$

Now let's show that each of these sets on the left hand side of this equation are disjoint. Suppose for a contradiction that there exists distinct $I_1, I_2 \subseteq J$ such that $H(I_1) \cap H(I_2)$ is non-empty. Assume without loss of generality that $I_2 \not\subseteq I_1$ and let $x \in H(I_1) \cap H(I_2)$. Now $x \in A_{I_1}$, $x \in A_{I_2}$ implies that $x \in A_{I_1 \cup I_2}$. But $I_1 \subsetneq I_1 \cup I_2$ by our assumption and this implies that $x \in \bigcup_{I_1 \subsetneq J} A_J$ and so

$$x \notin H(I_1)$$

which is a contradiction. So we have that $H(I_1)$ and $H(I_2)$ are disjoint for every distinct $I_1, I_2 \subseteq N$. Hence $\{H(K) : K \supseteq J\}$ is a partition. □

References

[1] Alan Tucker 1943 July 6-. *Applied combinatorics*. 6th. ID: $44IMP_ALMA_DS2155932560001591$. Hoboken, NJ: John Wiley and Sons, 2012.

[2] Victor Adamchik. *2005*. Carnegie Mellon University. URL: https://www.cs.cmu.edu/~adamchik/21-127/lectures/graphs_5_print.pdf.

[3] Professore Associato. *The matching problem*. Università degli Studi di Ancona. URL: http://www.dipmat.univpm.it/~demeio/Alabama_PDF/12.%20Finite_Sampling_Models/Matching.pdf.

[4] Muhammad Badar and Ansir Iqbal. *Polya's Enumeration Theorem*. Linnaeus University. 2010. URL: http://www.diva-portal.org/smash/get/diva2:324594/FULLTEXT01.pdf.

[5] E. A. Bender and J. R. Goldman. "On the Applications of Mobius Inversion in Combinatorial Analysis". In: *The American Mathematical Monthly* 82.8 (1975), pp. 789–803. DOI: 10.2307/2319793. URL: http://www.jstor.org/stable/2319793.

[6] Richard A. Brualdi. *Introductory Combinatorics*. 4th. ID: $44\text{IMP}_ALMA_DS2149286550001591$. Upper Saddle River, NJ: Pearson/Prentice Hall, 2004.

[7] David M. Burton. *Elementary Number Theory*. 6th. Singapore: McGraw-Hill Education (Asia), 2007.

[8] Peter J. Cameron. *Derangements*. University of St Andrews. 18 April 2013. URL: http://www-groups.mcs.st-andrews.ac.uk/~pjc/talks/pmc/d1.pdf.

[9] Dwight W. Chapman. *The Statistics of the Method of Correct Matchings*. Harvard university. URL: https://www.jstor.org/stable/1416561?read-now=1&loggedin=true&seq=1#page_scan_tab_contents.

[10] Brian Corney and Tom Davis. *Derangement*. Silicon Graphics. URL: http://www.geometer.org/mathcircles/.

[11] Dr Noach Dana-Picard. *Inclusion-Exclusion Principle*. Jerusalem College of Technology. 10 June 2002. URL: http://ndp.jct.ac.il/tutorials/discrete/node102.html.

[12] Isabel K. Darcy. *Derangement*. University of Iowa. URL: http://homepage.divms.uiowa.edu/~idarcy/COURSES/150/OLD/6_3bs.pdf.

[13] Sajal K. Das and Narsingh Deo. "Rencontres Graphs: a Family of Bipartite Graphs*". In: 1987.

[14] Daniel C. Fielder. "A generator of rook polynomials". In: *MATHEMATICA JOURNAL*. 9.2 (2004), pp. 371–375.

[15] Aaron Greicius. *Euler's Phi Function*. Loyola University Chicago. 14 Nov 2012. URL: http://gauss.math.luc.edu/greicius/Math201/Fall2012/Lectures/euler-phi.article.pdf.

[16] G. H. Hardy and E. M. Wright. *An Introduction to The Theory of Numbers*. 5th. United States, New York: Oxford Science Publications, 1979.

[17] Bruce Ikenaga. *The Euler Phi Function*. China University of Science and Technology. 31 July 2006. URL: http://ae.hc.cust.edu.tw/new_website/attachments/article/244/Lecture%207_Multiplicative%20Functions.pdf.

[18] J.H. van Lint and R.M.Wilson. *A Course in Combinatorics*. 2nd. United States of America: Cambridge University Press, 2001.

[19] L. Lovász. "I - Problems". In: (1993), pp. 15–107. DOI: https://doi.org/10.1016/B978-0-444-81504-0.50006-0. URL: http://www.sciencedirect.com/science/article/pii/B9780444815040500060.

[20] Ben Lynn. *Möbius Inversion*. [Accessed from 10 June 2018]. URL: https://crypto.stanford.edu/pbc/notes/numbertheory/mobius.html.

[21] Abigail G. Mitchell. "A block decomposition algorithm for computing rook polynomials". In: *arXiv preprint math/0407004* (2004).

[22] Ivan Niven, Herbert S. Zuckerman, and Hugh L. Montgomery. *An Introduction to The Theory of Numbers*. 5th. United States, Canada: John Wiley & Sons, Inc., 1991.

[23] Prof. Nige Ray. *The Principle of Inclusion/Exclusion*. Manchester University. URL: http://www.maths.manchester.ac.uk/~mrm/Teaching/DiscreteMaths/LectureNotes/InclusionExclusion.pdf.

[24] John Riordan. *An introduction to combinatorial analysis*. ID: $44IMP_ALMA_DS2138923220001591$. New York : London: Wiley; Chapman & Hall, 1958.

[25] Gian-Carlo Rota. "On the Foundations of Combinatorial Theory". In: ed. by Ira Gessel and Gian-Carlo Rota. Classic Papers in Combinatorics. ID: Rota1987. Boston, MA: Birkhäuser Boston, 1987, pp. 332–360. ISBN: 978-0-8176-4842-8. DOI: 10.1007/978-0-8176-4842-8_25. URL: https://doi.org/10.1007/978-0-8176-4842-8_25.

[26] Zvezdelina Stankova-Frenkel. *Möbius Inversion Formula, Multiplicative Functions*. UC Berkeley. 1998-99. URL: https://math.berkeley.edu/~stankova/MathCircle/Multiplicative.pdf.

[27] Erin Stuhlsatz. "Mobius Inversion Formula". In: *Unpublished* (2015). URL: https://www.whitman.edu/Documents/Academics/Mathematics/stuhlsatz.pdf.

[28] William T. Trotter. *Combinatorics and partially ordered sets : dimension theory*. Baltimore; London: Johns Hopkins University Press, 1992.

[29] Clay S. Turner. *Euler's Totient Function and Public Key Cryptography*. 7 Nov 2008. URL: http://www.claysturner.com/dsp/totient.pdf.