Queen's University
School of Computing

CISC 203: Discrete Mathematics for Computing II
Lecture 2: Combinatorics
Winter 2019

1 Basic Combinatorial Techniques

Humans have been counting things presumably ever since we gained the ability to recognize our fingers and toes. While it is difficult to ascertain exactly when humans developed the notion of counting, archaeologists and historians have found ancient tally sticks, used to record numbers, that are approximately 25 000–35 000 years old. Number systems are, by comparison, much more recent; Egyptians used a base 10 number system as early as 3100 BC, negative numbers were studied by the Chinese around 100–50 BC, and the earliest occurrence of the number zero was in 628 AD in a book by the Indian mathematician Brahmagupta.

Of course, counting things directly can be difficult or tedious at times, so mathematicians came up with an entire field of study devoted to counting objects, showing the existence of certain arrangements of objects, and constructing such arrangements. This field is called **combinatorics**. In this lecture, we will investigate a number of combinatorial techniques, discover connections between combinatorics and other aspects of discrete mathematics, and explore how combinatorics applies to computing.

1.1 Product Rule

Imagine you're making an important choice, like which CISC course to take as an elective next year. While planning your schedule, you realize you also want to take a mathematics elective next year. There are 5 CISC courses and 4 MATH courses from which you can choose your electives. How many possible pairs of courses exist?

The **product rule** tells you exactly how many possibilities exist in this scenario: if there are i ways to make one choice and j ways to make another choice, then there is a total of ij ways to make both choices. Generalizing from two choices to m choices, we get the product rule.

From our previous scenario, the product rule tells us that you have $5 \cdot 4 = 20$ different elective options available. Aside from scheduling, the product rule appears in many other areas both of computing and of life.

Example 1. Recall the set of binary digits $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$. A bit string is a concatenation of binary digits, and a byte is a bit string of length 8.

How many values can we store in one byte? Since a byte consists of eight binary digits, and since the set of binary digits contains two elements, we can store $2^8 = 256$ unique values in one byte.

Example 2. Car licence plates in Ontario follow a certain pattern: four uppercase letters followed by three numbers.

How many possible licence plates can the government produce? Let the set of letters consist of the usual 26 elements $\{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$, and let the set of numbers consist of the 10 single digits $\{0, 1, \ldots, 9\}$. Then, taking four elements from the letter set and three elements from the number set, we get a total of $26^4 \cdot 10^3 = 456\,976\,000$ possible licence plates.

As the previous examples might have revealed, the product rule is actually a disguised result about the Cartesian product of sets. Consider, for example, two sets $\{a, b, c\}$ and $\{d, e\}$. How many ways can we choose one element from each of these sets? If we take the Cartesian product of both sets, the resultant set gives us every possible pair of elements: $\{(a, d), (b, d), (c, d), (a, e), (b, e), (c, e)\}$. Therefore, we have six ways to choose one element from each set.

Thinking about the product rule in the context of set theory brings about the following formula.

Proposition 3 (Product rule). *Let $A_1, A_2, \ldots, A_m$ be disjoint sets each of finite cardinality. Then*

$$|A_1 \times A_2 \times \cdots \times A_m| = |A_1| \cdot |A_2| \cdot \quad \cdots \quad \cdot |A_m|.$$

Note that the sets $A_1, A_2, \ldots, A_m$ need not be disjoint, since our choices from one set don't depend on any of our previous or future choices from other sets.

1.2 Sum Rule

Imagine again that you're making an important choice. This time, you're deciding on an undergraduate honours project. Two professors in the School of Computing are interested in supervising you; the first professor has three project ideas and the second professor has four project ideas. How many options do you have for choosing your project?

In this scenario, we can't use the product rule because you may only choose a single project; you aren't choosing one project from each of the two professors. Instead, this is where the **sum rule** applies: if there are i ways to make one choice and j ways to make another choice, and the two choices cannot be made simultaneously, then there is a total of $i + j$ ways to make one of these choices. Generalizing from two choices to m choices, we get the sum rule.

From our previous scenario, the sum rule tells us that you have $3 + 4 = 7$ different project options available.

Example 4. Queen's University has a password policy similar to the following: a password must be between 10 and 12 characters long; it must consist of uppercase letters (A–Z), lowercase letters (a–z), and numbers (0–9); and it must contain at least one of each of those characters.

How many passwords exist that meet all of the above criteria? We can determine this using both the sum rule and the product rule. Let P_{10}, P_{11}, and P_{12} denote the sets of valid passwords of length 10, 11, and 12, respectively. The sum rule tells us that the total number of valid passwords is $P_{10} + P_{11} + P_{12}$.

Now, let's determine the values of P_{10}, P_{11}, and P_{12}. We do this using the product rule. First, we will determine all passwords of length n (which gives us 62^n), and then we will subtract from this value all passwords that don't contain at least one uppercase letter (36^n), lowercase letter (36^n), or number (52^n). This gives us the following values:

$$\begin{aligned} P_{10} &= 62^{10} - 36^{10} - 36^{10} - 52^{10} = 687\,431\,943\,039\,157\,248, \\ P_{11} &= 62^{11} - 36^{11} - 36^{11} - 52^{11} = 44\,256\,451\,766\,801\,594\,368, \text{ and} \\ P_{12} &= 62^{12} - 36^{12} - 36^{12} - 52^{12} = 2\,825\,912\,993\,235\,006\,394\,368. \end{aligned}$$

Therefore, there exists a total of 2 870 856 876 944 847 145 984 (2 sextillion) valid passwords. To put this number into context, astronomers estimate there are about 1 sextillion stars in the universe!

Again, just like with the product rule, the sum rule is a set theory result in disguise. This time, since we aren't taking elements from every set, we don't care about tuples of elements. Instead, we are considering all sets together—that is, the union of all sets—and selecting our element from the lot, so we just care about the total number of elements. From this observation, we get the following formula.

Proposition 5 (Sum rule). *Let $A_1, A_2, \ldots, A_m$ be disjoint sets each of finite cardinality. Then*

$$|A_1 \cup A_2 \cup \cdots \cup A_m| = |A_1| + |A_2| + \cdots + |A_m|.$$

1.3 Inclusion-Exclusion Principle

In our definitions of the product rule and the sum rule, we made one crucial assumption: that each of our sets $A_1, A_2, \ldots, A_m$ were disjoint. This assumption did not matter for the product rule, as we noted. Sadly, the sum rule is dependent on such an assumption. If any of our sets are not disjoint with any other set, then we cannot apply the sum rule as formulated in Proposition 5.

Luckily, we can make a small change to the sum rule to make the same idea work for sets which are not all disjoint. All we have to do is adjust the way we count to exclude elements that we may have counted more than once. We call this method of including individual elements and excluding common elements the **inclusion-exclusion principle**.

Example 6. Consider the sets $A = \{1, 2, 4, 8\}$ and $B = \{2, 4, 6, 8\}$. How many elements are in the union of the two sets, $A \cup B$?

Clearly, $|A \cup B| \neq 8$, since we would be counting the elements 2, 4, and 8 each twice. (In a set, we do not permit multiple copies of an element.)

What we can do instead is count the number of elements in each set and sum them together as usual, but also subtract from this sum the number of elements common to both sets. Since A and B share three elements, we have that $|A \cup B| = 4 + 4 - 3 = 5$. A quick check reveals that, indeed, $A \cup B = \{1, 2, 4, 6, 8\}$.

The previous example illustrates the inclusion-exclusion principle on two sets. Using set notation, the principle in this case would be written as

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

You might think this generalizes easily to three sets by adding $|C|$ to the sum and subtracting both $|A \cap C|$ and $|B \cap C|$ from the sum. However, this leads to another issue: doing so would have us subtract more than once any elements common to each of A, B, and C. Instead of overcounting, we're undercounting!

Since we're undercounting by exactly the number of elements common to all three sets, we must add that value back to the sum. As a result, the inclusion-exclusion principle on three sets is written thus:

$$|A \cup B \cup C| = |A| + |B| + |C| - |A \cap B| - |A \cap C| - |B \cap C| + |A \cap B \cap C|.$$

At this point, you might notice a pattern developing with each formulation of the inclusion-exclusion principle. We add cardinalities of single sets, we subtract cardinalities of pairs of sets, we add cardinalities of triples of sets, and so on. In general, when we introduce a term that involves k sets, we add if k is odd and we subtract if k is even.

Using this observation, let's write out the general form of the inclusion-exclusion principle.

Theorem 7 (Inclusion-exclusion principle)**.** *Let $A_1, A_2, \ldots, A_m$ be sets each of finite cardinality. Then*

$$\left|\bigcup_{i=1}^{m} A_i\right| = \sum_{k=1}^{m} (-1)^{k+1} \left(\sum_{1 \leq i_1 < \cdots < i_k \leq m} |A_{i_1} \cap \cdots \cap A_{i_k}| \right).$$

Proof. Omitted. □

The previous expression might seem a bit intimidating, so let's break it down into parts. The part of the expression to the left of the equals sign, $|\bigcup_{i=1}^{m} A_i|$, is a shorthand way to say $|A_1 \cup \cdots \cup A_m|$. To the right of the equals sign, we have two parts to consider. The sum, $\sum_{k=1}^{m}(-1)^{k+1}$, alternates between adding and subtracting the kth term of the expression; we add the term if k is odd and subtract the term if k is even. The term we are adding or subtracting, $\left(\sum_{1 \leq i_1 < \cdots < i_k \leq m} |A_{i_1} \cap \cdots \cap A_{i_k}|\right)$, is a notation-heavy way of writing "add together the cardinalities of all possible intersections of k sets for $1 \leq k \leq m$".

If you're worried about having to remember the unwieldy general form of the inclusion-exclusion principle, don't be; most of the time, you will only need to use the cleaner two- or three-set formula. But, as an exercise, try figuring out the statement of the inclusion-exclusion principle on four sets.

1.4 Pigeonhole Principle

In a post office or mailroom, mail is sorted into small slots in the wall, colloquially called "pigeonholes". (This word wasn't just made up for fun; pigeon holes were originally boxes on walls used to store domesticated birds.) In 1834, a German mathematician named Peter Gustav Lejeune Dirichlet studied combinatorial problems involving n items being placed into m containers where $n > m$. Dirichlet needed a name for his work; his father happened to be a postmaster, and postmasters place mail into pigeonholes, so that term seemed quite fitting. From this, the pigeonhole principle was coined.

Theorem 8 (Pigeonhole principle)**.** *If n elements are partitioned into m subsets, then at least one subset must contain at least $\lceil n/m \rceil$ elements.*

Proof. Suppose we have n elements partitioned into m subsets. By contradiction, suppose that no subset contains more than $\lceil n/m \rceil - 1$ elements. Then the total number of elements is at most

$$m\left(\left\lceil \frac{n}{m} \right\rceil - 1\right) < m\left(\left(\frac{n}{m} + 1\right) - 1\right) = n.$$

Thus, we must have fewer than n elements. However, we assumed we had exactly n elements. Therefore, our assumption was incorrect and at least one subset must contain at least $\lceil n/m \rceil$ elements. □

To illustrate the pigeonhole principle, let's consider an example that involves (what else?) pigeons. Suppose there are nine nests in a tree, and ten pigeons fly to that tree to roost. Nine pigeons can each take one nest, but that leaves one pigeon without a nest. If all ten pigeons are in a nest, then it must be the case that at least one nest contains more than one pigeon.

So what do pigeons and nests have to do with combinatorics? Well, nothing. But the idea behind the pigeonhole principle can be applied to a number of mathematical and computational problems.

Example 9. In the 2017–2018 academic year, there were 11 783 students enrolled in the Faculty of Arts and Science at Queen's University. Assume that each of these students was born in the same four-year period of 1996–1999. The given four-year period consisted of 1 461 days. Therefore, by the pigeonhole principle, at least $\lceil 11\,783/1\,461 \rceil = 9$ students in the Faculty of Arts and Science were born on the same day.

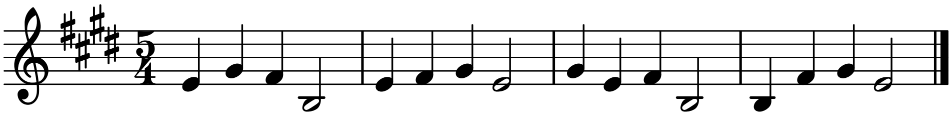
Example 10. Lossless compression algorithms allow users to compress files without losing any of the original data. In a perfect world, a lossless compression algorithm would always make files smaller. However, on some files, the algorithm actually increases the file size. Why?

By contradiction, suppose there exists a lossless compression algorithm that compresses every file F into a compressed file F', where $\text{size}(F') \leq \text{size}(F)$. Let M be the smallest natural number such that there exists a file F with $\text{size}(F) = M$ that can be compressed to a file F' with $\text{size}(F') = N$.

Since $N < M$, every size-N file keeps its original size after compression. There are 2^N such files. Since F' is also of size N, we have a total of $2^N + 1$ files of size N. However, $2^N < 2^N + 1$, so the pigeonhole principle tells us that two different compressed files of size N must be identical. Such files cannot be losslessly decompressed, since we don't know which of the two original files they once were. Therefore, our assumption was incorrect and no lossless compression algorithm can always make a file smaller.

2 Permutations

If you've ever found yourself in the downtown area around the top of each hour, you would have heard a tune familiar to most Kingstonians. It goes like this:



This tune is called the *Westminster Quarters*. It is very commonly heard ringing out of clock towers or church steeples to indicate that the previous hour has passed and that a new hour is beginning.

The *Westminster Quarters* is a tune with an unusual property: it consists only of four notes. For those of you unfamiliar with sheet music, these notes are B_3, E_4, $F\sharp_4$, and $G\sharp_4$. Each of the four bars of the tune plays these notes in a different order; for instance, the first bar plays the notes in the order E_4, $G\sharp_4$, $F\sharp_4$, B_3.

Mathematically speaking, we say that each bar of the tune plays a different **permutation** of the four notes. Indeed, this is how we will (informally) define a permutation: an arrangement of elements from a given set into a specific order.

2.1 Definition

To define the notion of a permutation formally, we require some set theory and some relation theory.

Definition 11 (Permutation)**.** Given a set A, a permutation of A is a bijective function $\pi : A \to A$.

That's it! A permutation is just a function that maps elements of a set A to elements of the same set A. In other words, if our informal definition of a permutation is an arrangement of elements, then the permutation π itself is what performs the rearranging of elements.

Example 12. Suppose we have a set $A = \{1, 2, 3, 4, 5, 6\}$, and suppose further that we have a permutation π defined as

$$\begin{aligned} \pi(1) &= 3; & \pi(4) &= 2; \\ \pi(2) &= 4; & \pi(5) &= 1; \\ \pi(3) &= 5; & \pi(6) &= 6. \end{aligned}$$

Then $\pi(A) = \{3, 4, 5, 2, 1, 6\}$, $\pi(\pi(A)) = \{5, 2, 1, 4, 3, 6\}$, and so on.

Observe here that, even though each of A, $\pi(A)$, and $\pi(\pi(A))$ contain the same elements, we consider them to be different permutations of the set A. This is an important fact to remember: ordering matters for permutations. (The scenario is different with combinations, as we will see in the next section.)

We can more compactly represent the permutation π from the previous example using certain notations. The **two-line notation** uses, as you might expect, a two-line matrix; the first line lists the elements of the set A, and the second line lists the permuted elements $\pi(A)$. Thus,

$$\pi = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 2 & 1 & 6 \end{pmatrix}.$$

Alternatively, we could represent the permutation π using **cycle notation**. In this notation, we choose a starting element and follow a "cycle" through the permutation until we arrive back at the starting element. We perform these steps until we have written each element in A. Thus, $\pi = (1\ 3\ 5)(2\ 4)(6)$.

Now, let's get back to the topic of these lecture notes: counting. Given a set A with n elements, how many permutations of A exist? Let's investigate by way of example.

Example 13. Suppose $m = 4$, so $A = \{a_1, a_2, a_3, a_4\}$. Let π be a permutation of A. The permutation π is essentially taking each element of A and inserting it into a blank spot of $\pi(A)$.

Consider the element a_1. The current state of $\pi(A)$ is $\{\llcorner\ \llcorner\ \llcorner\ \llcorner\}$, so there are four spaces into which a_1 can be inserted. Suppose a_1 is inserted into the second space.

Consider the element a_2. The current state of $\pi(A)$ is $\{\textvisiblespace\ a_1\ \textvisiblespace\ \textvisiblespace\}$, so there are three spaces into which a_2 can be inserted. Suppose a_2 is inserted into the fourth space.

Consider the element a_3. The current state of $\pi(A)$ is $\{\textvisiblespace\ a_1\ \textvisiblespace\ a_2\}$, so there are two spaces into which a_3 can be inserted. Suppose a_3 is inserted into the first space.

Finally, consider the element a_4. The current state of $\pi(A)$ is $\{a_3\ a_1\ \textvisiblespace\ a_2\}$, so there is only a single space into which a_4 can be inserted.

Altogether, there are $4 \times 3 \times 2 \times 1 = 24$ different ways to arrange the elements of A, so there are 24 possible permutations of A.

In general, the number of permutations of a set with n elements is equal to the product of every natural number from 1 to n. We can represent this product more concisely using the **factorial** notation: we write $n!$ to denote the product $1 \times 2 \times \cdots \times n$. Using this notation, we can express the formula to find the number of permutations of an n-element set.

Theorem 14. *The number of permutations of a set with n distinct elements is $P(n, n) = n!$*

Proof. Omitted. (See the proof of Theorem 16 for the general case.) □

Remark. The value $n!$ grows very quickly as n increases. We can compare the growth rate of $n!$ to that of a function involving only n and constants using **Stirling's approximation**. For large-enough n, we have

$$n! \approx \sqrt{2\pi n}\left(\frac{n}{\mathrm{e}}\right)^n.$$

You might have noticed that Theorem 14 used the notation $P(n, n)$ to represent the number of permutations of an n-element set. Why did we write n twice? The first n in our notation denotes the number of elements in our original set, while the second n denotes the number of elements we are permuting from that set. Up until now, we have always permuted every element in the set (hence, why we wrote the second n), but we could just as easily consider permutations of subsets of elements.

Since this new idea of permuting subsets of elements is not, strictly speaking, a permutation in the sense of Definition 11, we should give it a new name. Thus, we will say that a ***k*-permutation** is a permutation of k elements taken from an n-element set, where $0 \leq k \leq n$.

Example 15. Consider again the set $A = \{1, 2, 3, 4\}$. Some possible 2-permutations of A are $\{1, 3\}$, $\{1, 2\}$, $\{2, 1\}$, $\{2, 4\}$, $\{3, 1\}$, and $\{4, 2\}$.

Obviously, the 2-permutations of A listed in the previous example are not all that are possible. How many 2-permutations—or, more generally, how many k-permutations—exist for an n-element set? We can use the same ideas as before to determine this value, except this time we need not take all n elements from our set.

Theorem 16. *The number of k-permutations of a set with n distinct elements, where $0 \leq k \leq n$, is*

$$P(n, k) = \frac{n!}{(n-k)!}.$$

Proof. For all i where $0 \leq i \leq (k-1)$, there are $n - i$ ways to choose the ith element of the permutation. By the product rule, we have that

$$\begin{aligned} P(n, k) &= n(n-1)(n-2)\cdots(n-(k-1)) \\ &= n(n-1)(n-2)\cdots(n-k+1). \end{aligned}$$

We can rewrite the right-hand side of the above expression as

$$\begin{aligned} n(n-1)(n-2)\cdots(n-k+1) &= \frac{n(n-1)(n-2)\cdots(n-k+1)(n-k)(n-k-1)\cdots(2)(1)}{(n-k)(n-k-1)\cdots(2)(1)} \\ &= \frac{n!}{(n-k)!}. \end{aligned}$$

□

Note that, as a consequence of Theorem 16, we have $P(n, 0) = 1$ and $P(n, 1) = n$ for all $n \geq 0$. We also have $P(n, n) = n!$ for all $n \geq 0$, meaning that Theorem 14 is really just a corollary of Theorem 16.

Remark. From Definition 11, we know that a permutation is a bijective function. From this, we can conclude that the value $P(n, n)$ counts the number of bijective functions from a set of size n to a set of size n. What, then, does the value $P(n, k)$ count? If $k < n$, then we can't have a bijection (since we can't have a surjection). Because we're mapping each of the n elements in the range to at most one of the k elements in the domain, $P(n, k)$ is counting the number of injective functions from a set of size k to a set of size n.

Example 17. A certain instructor is creating a midterm exam for his discrete mathematics class. Suppose he has a question bank containing 50 different questions. If the midterm exam consists of 5 questions, then the instructor can create a total of $P(50, 5) = \frac{50!}{(50-5)!} = 254\,251\,200$ midterm exams.

Example 18. The Examinations Office is scheduling three CISC exams and three MATH exams, all of which are different. Three exams can be held per day, and the office wants to schedule all CISC exams on one day and all MATH exams on the other day. Under such constraints, the office has (3!)(3!)(2!) ways to schedule all exams: 3! ways of scheduling three CISC exams, 3! ways of scheduling three MATH exams, and 2! ways to arrange the "CISC exam" day and the "MATH exam" day.

Our final example illustrates one of the most famous applications of permutations to computing.

Example 19. The traveling salesperson problem is stated as follows: "Given a list of n cities and a list of distances between each pair of cities, what is the shortest route that both visits all cities and ends in the origin city?"

The traveling salesperson problem is well-known for being a difficult problem for computers to solve. This is because, if we take a brute force approach to solving the problem, our solver must list all routes through each city, determine whether that route ends in the origin city, and then keep only the shortest route. Since we have n cities to visit, in the worst case this approach will require our solver to analyze $n!$ different routes.

For small values of n, this isn't bad; a brute force approach could easily solve the traveling salesperson problem for 3 or 4 cities. However, for larger n, the problem quickly grows out of hand. If we have a list of 20 cities, and if our solver checks 100 solutions per second, it would take 771 000 000 years to check every solution!

2.2 Permutations with Repetition

Earlier, we learned that there exist $n!$ permutations of an n-element set. We got the $n!$ term from the fact that, after selecting the ith element of the permutation, we had $(i - 1)$ elements remaining to permute. But what if we could select elements from the set more than once?

We already saw this idea in disguise earlier in these lecture notes; recall the example where we determined how many values can be stored in one byte. Since a byte is 8 bits, and since a bit can be either `0` or `1`, we were essentially constructing permutations by choosing elements from the set $\mathbb{B} = \{\mathtt{0}, \mathtt{1}\}$ more than once. In the example, we determined that it is possible to represent 2^8 values in one byte; in other words, 8 selections from a 2-element set.

Permutations with repetition, then, is just a specific application of the product rule where we're making k selections from k "copies" of the same set. (In reality, we only have one copy of the set, but it can be easier to illustrate the process by imagining multiple copies.)

Theorem 20. *The number of k-permutations of a set with n distinct elements, with repetition, is n^k.*

Proof. For all i where $0 \leq i \leq (k - 1)$, there are n ways to choose the ith element of the permutation, since repetition is allowed. By the product rule, we have that $\underbrace{n \times \cdots \times n}_{k \text{ times}} = n^k$. □

2.3 Permutations with Indistinguishable Elements

We now know quite a lot about permutations of sets, and we also know that sets contain only one copy of each distinct element. If we modify the definition of a set to allow for multiple copies of each element, then we get what is known as a **multiset**. For instance, $\{1, 2\}$ is a set, but $\{1, 2, 2\}$ is a multiset. Note that with multisets, just as with sets, order does not matter; the multisets $\{1, 2, 2\}$ and $\{2, 1, 2\}$ are the same multiset. What can we say about permutations of multisets?

We call repeated elements within a multiset **indistinguishable**. Since ordering matters with permutations, we must take care that two permutations that look identical (due to the indistinguishable elements) are not both counted as separate permutations.

In order to count the number of permutations of a set with indistinguishable objects, we begin by counting the number of permutations of the set as usual. We then account for the overcounting of identical permutations by dividing our total permutations by the number of ways we can permute just the indistinguishable elements.

Example 21. The canonical example of a word with indistinguishable letters is MISSISSIPPI. In this word, we have four I's, four S's, and two P's, all of which are indistinguishable. We can "distinguish" the letters for illustrative purposes by subscripting them:

$$\mathrm{M\ I_1\ S_1\ S_2\ I_2\ S_3\ S_4\ I_3\ P_1\ P_2\ I_4}.$$

If we permute two or more indistinguishable letters, then we should not get two or more different permutations. For instance, MISSISSIP$_1$P$_2$I and MISSISSIP$_2$P$_1$I should be considered the same permutation.

We have a total of 11! ways to permute the word MISSISSIPPI. We then divide that by 4!, 4!, and 2! to account for the indistinguishable letters I, S, and P, respectively, which gives us a total of $\frac{11!}{(4!)(4!)(2!)}$ permutations.

If we consider each indistinguishable element to be in its own "class", we can formulate this counting technique as follows.

Theorem 22. *The number of permutations of a set with n elements, where n_i of the elements are in class i for $1 \leq i \leq r$, is*

$$\frac{n!}{n_1!n_2!\cdots n_r!}$$

Proof. Suppose we have n_1 elements in class 1 and we have n positions in which to insert these elements. We can insert all n_1 elements in $P(n, n_1)$ ways, but ordering doesn't matter since the n_1 elements are indistinguishable, so we must divide by $P(n_1, n_1)$ to compensate for overcounting. Thus, we have

$$\begin{aligned}\frac{P(n, n_1)}{P(n_1, n_1)} &= \frac{n!/(n-n_1)!}{n_1!/(n_1-n_1)!}\\ &= \frac{n!}{n_1!(n-n_1)!}\end{aligned}$$

ways to insert the n_1 elements in class 1, and we have $(n - n_1)$ positions remaining.

If we perform the same steps for every class n_i, where $1 \leq i \leq r$, then by the product rule we have that

$$\frac{n!}{n_1!(n-n_1)!} \cdot \frac{(n-n_1)!}{n_2!(n-n_1-n_2)!} \cdot \ \cdots \ \cdot \frac{(n-n_1-\cdots-n_{r-1})!}{n_r!0!} = \frac{n!}{n_1!n_2!\cdots n_r!}. \qquad \square$$

Example 23. The Examinations Office is booking 12 rooms for exams to be written during one week in April. In this week, five CISC exams will be written (three of which are different sections of the same course), four MATH exams will be written (two of which are different sections of the same course), and three PHYS exams will be written (all of which are different courses). Different sections of the same course write the same exam, so for room-booking purposes each section is indistinguishable.

There are 12! ways to assign exams to rooms, but the office doesn't care in which room different sections write the same exam. Therefore, they must divide by the number of different sections writing the same exam. This gives a total of $\frac{12!}{(3!)(2!)}$ ways to schedule exams for the week.