Counting is the the task of finding the number of elements (also called the *cardinality*) of a given set. When the set is small, we can count its elements "by hand". When sets are larger we need a more systematic way to count.

Say you have a six-sided die and a two-sided coin — it comes out heads (H) or tails (T). What is the number of possible outcomes when both the die and the coin are tossed? There are 6 possible outcomes for the die and 2 for the coin, so the total number of outcomes is $6 \times 2 = 12$.

It will be useful to describe this kind of problem using the language of sets. The set S of possible outcomes of the die is $S = \{1, 2, 3, 4, 5, 6\}$, so $|S| = 6$. The set T of possible outcomes of the coin is $T = \{\mathtt{H}, \mathtt{T}\}$, so $|T| = 2$. The set of possible outcomes of the die *and* the coin is the *product set* $S \times T$:

$$S \times T = \{(1, \mathtt{H}), (1, \mathtt{T}), (2, \mathtt{H}), (2, \mathtt{T}), \dots, (6, \mathtt{H}), (6, \mathtt{T})\}.$$

The number of elements of $S \times T$ is $|S| \cdot |T| = 6 \cdot 2 = 12$.

In general, given any two finite sets S and T the *product set* $S \times T$ consists of all ordered pairs of elements (s, t) such that s is in S and t is in T:

$$S \times T = \{(s, t) \colon s \in S \text{ AND } t \in T\}.$$

The number of elements of $S \times T$ is the product of the number of elements of S and the number of elements of T, i.e., $|S \times T| = |S| \cdot |T|$.

Let's do another example: Let R and B be the sets of outcomes of a toss of a red and a blue six-sided die, respectively. Then $R = \{1, 2, 3, 4, 5, 6\}$ and $B = \{1, 2, 3, 4, 5, 6\}$. When both dies are tossed, the set of outcomes is

$$R \times B = \{(1, 1), (1, 2), \dots, (6, 6)\}$$

and the number of outcomes is $|R \times B| = |R| \cdot |B| = 36$.

In cases like this when $S = T$ we can denote the set $S \times T$ by S^2. (This is the square of a set, not the square of a number). The set S^2 has $|S|^2$ elements.

We can also take the product of more than two sets. The set $S_1 \times \cdots \times S_n$ (where $S_1, \dots, S_n$ are finite sets) consists of all *sequences*[1] $(s_1, \dots, s_n)$ where s_i is in S_i for all i between 1 and n:

$$S_1 \times \cdots \times S_n = \{(s_1, \dots, s_n) \colon s_1 \in S_1 \text{ AND } \dots \text{ AND } s_n \in S_n\}.$$

The set $S_1 \times \cdots \times S_n$ has $|S_1| \cdots |S_n|$ elements.

When $S_1 = \cdots = S_n = S$, we write S^n for $S_1 \times \cdots \times S_n$. This set has $|S|^n$ elements.

For example, the set of outcomes when 9 different six-sided dies are tossed is $\{1, 2, 3, 4, 5, 6\}^9$. This set has 6^9 elements, so there are 6^9 possible outcomes.

1 Functions, bijections, and counting

One technique for counting the number of elements of a set S is to come up with a "nice" correspondence between a set S and another set T whose cardinality we already know.

[1] The order of elements in a sequence matters and there can be repetitions: For example, $(1, 1, 2)$, $(2, 1, 1)$, and $(1, 2, 2)$ are all different sequences.

Let's count the number of subsets of the set $S_n = \{1, 2, 3, \ldots, n\}$.

For example, when $n = 1$, $S_1 = \{1\}$ are there are two subsets: $\varnothing$ and $\{1\}$. When $n = 2$, $S_2 = \{1, 2\}$ and there are four subsets: $\varnothing$, $\{1\}$, $\{2\}$, and $\{1, 2\}$.

I suspect that most of you can come up with a proof by induction that shows the set S_n has 2^n elements. Instead of using induction, let me show you another way.

The set $\{0, 1\}^n$ consists of all possible n-bit sequences. This is a product set, so it has 2^n elements. We will show how to represent subsets of S_n by elements of $\{0, 1\}^n$ in a unique way: Each subset of S_n is represented by an n bit sequence, and each n bit sequence represents some subset. So their number must be the same.

Here is how the representation works: The subset $\{s_1, \ldots, s_k\}$ of S_n is represented by the bit sequence that has ones in positions $s_1, \ldots, s_k$ and zeros in all the other positions. For example, if $n = 7$, the subset $\{3, 4, 6\}$ of S_n is represented by the bit sequence $(0, 0, 1, 1, 0, 1, 0)$ in $\{0, 1\}^n$.

Clearly every subset of S_n is represented by some n bit sequence. It is also true that every n bit sequence represents a subset: The sequence $(b_1, \ldots, b_n)$ represents the set of all i between 1 and n such that $b_i = 1$. For example, the sequence $(0, 0, 1, 1, 0, 1, 1)$ represents the set $\{3, 4, 6, 7\}$. We showed a "one-to-one" correspondence between the subsets of S_n and the elements of $\{0, 1\}^n$ (n-bit sequences), so their number must be the same.

As this is our first argument of this type, let us explicitly write out the correspondence between the elements of S_2 and the sequences in $\{0, 1\}^2$:

$$\varnothing \leftrightarrow (0, 0) \qquad \{1\} \leftrightarrow (1, 0) \qquad \{2\} \leftrightarrow (0, 1) \qquad \{1, 2\} \leftrightarrow (1, 1).$$

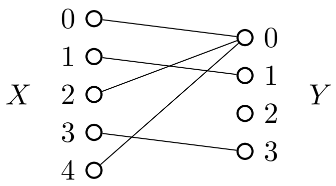
It is useful to have a language to describe these correspondences between sets. For this we need to talk about functions.

Functions A *function* f from a set X to a set Y associates to every element x in X an element $f(x)$ in Y. For example, $f(0) = 0$, $f(1) = 1$, $f(2) = 0$, $f(3) = 3$, $f(4) = 0$ is a function from set $X = \{0, 1, 2, 3, 4\}$ to the set $Y = \{0, 1, 2, 3\}$.

One way to specify a function is to list the values on all elements x in X:

x:	0	1	2	3	4
$f(x)$:	0	1	0	3	0

or by means of a bipartite graph G_f whose vertices[2] are partitioned into X and Y and whose edges are those $\{x, y\}$ such that $f(x) = y$:



A more convenient way is to describe the function using logic. This can be done in many ways, for

[2] We abuse notation here and use the same label for vertices in X and vertices in Y.

example by giving a formula for calculating f:

$$f(x) = \begin{cases} x, & \text{if } x \text{ is odd,} \\ 0, & \text{if } x \text{ is even} \end{cases} \qquad x \in \{0, 1, 2, 3, 4\}$$

an algorithm for calculating f:

Function $f(x)$, where $x \in \{0, 1, 2, 3, 4\}$:
    Let $b = x \bmod 2$.
    Output $b \cdot x$.

or, say, by giving a recurrence:

$$f(x) = (x \bmod 2) \cdot (f(x-2) + 2) \text{ for } x \geq 2, \qquad f(0) = 0, f(1) = 1.$$

No matter which way you describe a function, you must make sure that $f(x)$ is *well-defined* for every x in X: This value is specified and it is specified uniquely. For example, here is a *bad* way to define a function:

$$f(x)^2 = x.$$

this specification is consistent with both $f(1) = 1$ and $f(1) = -1$, so it does not uniquely describe the value $f(1)$.

We write $f\colon X \to Y$ for "a function from set X to set Y".

A function $f\colon X \to Y$ is a *injective* if distinct elements in x are mapped to distinct elements in Y. More precisely, f is injective if for every pair of elements x and x' in X such that $x \neq x'$, we have $f(x) \neq f(x')$. The above function is not injective because $0 \neq 2$ but $f(0) = f(2)$.

A function $f\colon X \to Y$ is *surjective* if every element y in Y is mapped to by some x in X. More precisely, f is surjective if for every y in Y there exists x in X such that $f(x) = y$. The above function is not surjective because $2 \in Y$ but nothing maps to it.

Let's assume X and Y are finite sets. When f is injective, the edges of the graph G_f are a matching and they match all the vertices of X, so Y must be at least as large as X, that is $|Y| \geq |X|$. When f is surjective, then every element of Y can be matched to some element of X (by taking an arbitrary edge incident to it), so Y can be at most as large as X, that is $|Y| \leq |X|$.

We say f is *bijective* if it is injective and surjective. When X, Y are finite and f is bijective, the edges of G_f form a perfect matching between X and Y, so $|X| = |Y|$.

This is why bijective functions are useful for counting: If we know $|X|$ and can come up with a bijective $f\colon X \to Y$, then we immediately get that $|Y| = |X|$. The tricky part is coming up with f and showing that it is bijective — namely, both injective and surjective.

Let's rework our previous example in this language. Let $X = \{0, 1\}^n$ and Y be the set of all subsets of $\{1, \ldots, n\}$. Let $f\colon X \to Y$ be the function that on input $(b_1, \ldots, b_n)$ outputs the set of indices i such that $b_i = 1$, namely

$$f((b_1, \ldots, b_n)) = \{i\colon b_i = 1\}.$$

Theorem 1. *f is a bijective function.*

Proof. First we prove f is injective: Different bit sequences map to different sets. Suppose two bit sequences x and x' are different. Then they must differ in one of their positions, say position i. Then the element i is in one of the sets $f(x), f(x')$ but not in the other, so $f(x) \neq f(x')$.

Now we show f is surjective. Given any subset S of Y, we'll show there exists a bit sequence x such that $f(x) = S$. Let x be the n-bit sequence that has a 1 in position i if i is in S and 0 if i is not in S. Then $f(x) = S$. □

Corollary 2. *The number of subsets of* $\{1, \ldots, n\}$ *is* 2^n.

Proof. By Theorem 1, there is a bijection from elements of $\{0,1\}^n$ to subsets of $\{1, \ldots, n\}$. Therefore the number of subsets of $\{1, \ldots, n\}$ equals the number of elements of $\{0,1\}^n$, which is 2^n. □

2 The sum rule

The sum rule says that if $A_1, \ldots, A_n$ are *disjoint* sets then

$$|A_1 \cup \cdots \cup A_n| = |A_1| + \cdots + |A_n|.$$

This rule is useful when the set whose number of elements we want to count can be written as a disjoint union of simpler sets.

For example, if you have 10 red balls, 7 blue balls, and 4 red balls, then the total number of balls you have is $10 + 7 + 4 = 21$. You could have done this in first grade. So why do we need sets and unions? Sets help us break complicated counting problems into simpler ones.

Suppose you need to choose a password between 6 and 8 symbols out of which the first symbol must be a letter (lowercase or uppercase) and the remaining ones must be letters or digits. How many possible passwords are there?

Let P be the set of possible passwords, L be the set of letters, and S be the set of symbols (letters or digits). The English alphabet has 26 letters, each of which can be lowercase or uppercase, so $|L| = 52$. The set S contains all the letters plus the ten digits, so $|S| = 62$.

To count the number of passwords $|P|$, it makes sense to "decompose" the set P in terms of the simpler sets L and S. Here is how we do it. First, P is a disjoint union of six, seven, and eight letter passwords — let's denote these sets by P_6, P_7, and P_8. Therefore

$$|P| = |P_6| + |P_7| + |P_8|.$$

The set P_6 is a product set: It consists of all sequences of a letter followed by five symbols, so $P_6 = L \times S \times S \times S \times S \times S = L \times S^5$. Similarly, $P_7 = L \times S^6$ and $P_8 = L \times S_7$. Therefore

$$\begin{aligned} |P| &= |L \times S^5| + |L \times S^6| + |L \times S^7| \\ &= |L| \cdot |S|^5 + |L| \cdot |S|^6 + |L| \cdot |S|^7 \\ &= 52 \cdot 62^5 + 52 \cdot 62^6 + 52 \cdot 62^7 \\ &\approx 1.8 \cdot 10^{14}. \end{aligned}$$

The sum rule can also be used backwards as in the following example. You toss a blue six-sided die and a red six-sided die. How many outcomes are there in which the face values of the two dice come out different? Here, the set A of all possible outcomes for the pair of dice can be written as a disjoint union of the set D of outcomes in which the two face values are different and the set S in which the two are the same. By the sum rule, $|A| = |D| + |S|$. Since $|A| = 36$ and $|S| = 6$, we get that $|D| = 36 - 6 = 30$ possible outcomes in which the two face values are different.

3 The general product rule and permutations

Let's see a different solution to the last example. The set of possible outcomes in which the two face values are different is *not* a product set, but we can still reason like this: There are 6 possibilities for the face value of the first die. For each one of these six possibilities, we are interested in the outcomes in which the second die has a different face value. No matter what the toss of the first die came out to be, there are always five possible choices for the toss of the second die that make their face values different. Therefore the total number of outcomes is $6 \times 5 = 30$.

This is an instance of the *general product rule*: Given a set S of sequences of length k, where

- There are n_1 possible first entries,
- There are n_2 possible second entries for each first entry
- There are n_3 possible third entries for each combination of first and second entries, and so on up to n_k

the size of S is $n_1 \cdot n_2 \cdots n_k$.

A *permutation* of a set S is a sequence that contains every element of S exactly once. For example, the permutations of the set $\{1, 2, 3\}$ are the sequences $(1, 2, 3)$, $(1, 3, 2)$, $(2, 1, 3)$, $(2, 3, 1)$, $(3, 1, 2)$, $(3, 2, 1)$.

How many permutations does an n element set have? The first entry in the permutation sequence can be chosen in n possible ways. Each such choice leaves out $n - 1$ possibilities for the second element. Each combination of choices for the first two elements leaves out $n - 2$ combinations for the third element. Continuing like this, we get that the number of permutations of an n element set is

$$n \cdot (n - 1) \cdot (n - 2) \cdots 1 = n!.$$

Let's do another example of the product rule. In how many ways can you place three different pieces on an 8×8 chessboard — a bishop, a knight, and a pawn — so that no two pieces share a row or a column? The position of the three pieces is specified by a six numbers (br, bc, kr, kc, pr, pc). The first number br indicates the bishop's row, the second number bc indicates the bishop's column, and so on.

We can count the number of allowed sequences with the generalized product rule. There are 8 possibilities for the bishop's row br. For each such choice, there are 8 possibilities for the bishop's column bc. Once the bishop's position was chosen, there are 7 choices for the knight's row kr (any one but br) and (for each of them) 7 choices from kc (any one but bc. Once the bishop and the knight were positioned, there are 6 choices for pr and (for each of them) 6 choices of pc. The total number of configurations is $8 \cdot 8 \cdot 7 \cdot 7 \cdot 6 \cdot 6 = (8 \cdot 7 \cdot 6)^2$.

4 The pigeonhole principle

The pigeonhole principle says that if you toss more pigeons into fewer holes, at least two pigeons will land in the same hole. If X is the set of pigeons, Y is the set of holes, and $f\colon X \to Y$ is the function that tells you which pigeon goes into which hole, we get the following mathematical