**Material**

**Definition**

Alternative Title: combinatorial mathematics. Combinatorics, also called combinatorial mathematics, the field of mathematics concerned with problems of selection, arrangement, and operation within a finite or discrete system. Included is the closely related area of combinatorial geometry.

Noted : Combinatorics itself belongs to the "Discrete" family in mathematics. In middle and high school level learning, combinatorics is found in opportunity materials. If opportunity is a way to study the probability of a certain event occurring, then combinatorics is devoted to the way a certain set of objects is arranged.

*Kombinatorika sendiri tergolong ke dalam rumpun “Diskrit” dalam matematika. Dalam pembelajaran setingkat SMP dan SMA, kombinatorika ditemukan pada materi peluang. Jika peluang merupakan suatu cara untuk mempelajari kemungkinan terjadinya suatu peristiwa tertentu, maka kombinatorika dikhususkan pada cara penyusunan sekumpulan objek tertentu.*

**Used for**

Combinatorics has many applications in other areas of mathematics, including graph theory, coding and cryptography, and probability.

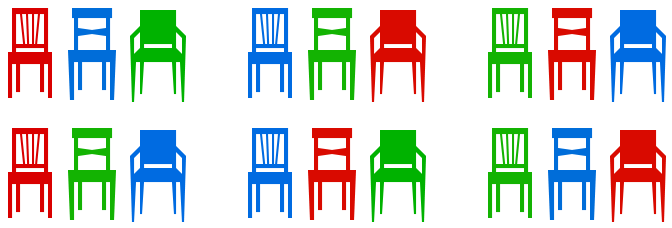
Combinatorics is especially useful in computer science. Combinatorics methods can be used to develop estimates about how many operations a computer algorithm will require. Combinatorics is also important for the study of discrete probability. Combinatorics methods can be used to count possible outcomes in a uniform probability experiment.

**Factorials**

Combinatorics can help us count the number of *orders* in which something can happen. Consider the following example:

*In a classroom there are 3 pupils and 3 chairs standing in a row. In how many different orders can the pupils sit on these chairs?*

Let us list the possibilities – in this example the *3* different pupils are represented by *3* different colours of the chairs.

**

There are *6* different possible orders. Notice that the number of possible orders increases very quickly as the number of pupils increases. With 6 pupils there are 720 different possibilities and it becomes impractical to list all of them. Instead we want a simple formula that tells us how many orders there are for *n* people to sit on *n* chairs. Then we can simply substitute 3, 4 or any other number for *n* to get the right answer.

Suppose we have *4* chairs and we want to place *four pupils* on them. *There are 4 pupils who could sit on the first chair. Then there are 3 pupils who could sit on the second chair. There are 2 choices for the third chair, and only one choice for the final chair.* In total, there are

possibilities. To simplify notation, mathematicians use a “!” called factorial. For example, 5! (“five factorial”) is the same as 5 × 4 × 3 × 2 × 1. Above we have just shown that there are *n*! possibilities to order *n* objects.

**EXERCISE**

**SOLUTION**



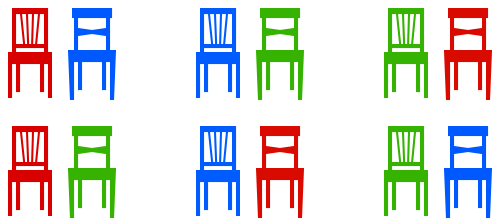
In how many different ways could 23 children sit on 23 chairs in a Maths Class? If you have 4 lessons a week and there are 52 weeks in a year, how many years does it take to get through all different possibilities? *Note: The age of the universe is about 14 billion years.*

**Permutations**

The method above required us to have the same number of pupils as chairs to sit on. But what happens if there are not enough chairs?

*How many different possibilities are there for any 2 of 3 pupils to sit on 2 chairs? Note that 1 will be left standing, which we don’t have to include when listing the possibilities.*

Let us start again by listing all possibilities:

**

To find a simple formula like the one above, we can think about it in a very similar way. *undefined* In total there are

possibilities. Again we should think about generalising this. We start like we would with factorials, but we stop before we reach 1. In fact we stop as soon as we reach the number students without chair. When placing **7** students on **3** chairs their are

7 × 6 × 5   =  7 × 6 × 5 × 4 × 3 × 2 × 14 × 3 × 2 × 1  =  **7**!4!  =  **7**!(**7** – **3**)!

possibilities, since the 4 × 3 × 2 × 1 will cancel each other. Again there is a simpler notation for this: **7P3**. If we want to place *n* objects in *m* positions there are

*n***P***m*   =  *n*!(*n* – *m*)!

possibilities. The **P** stands for “**p**ermutations”, since we are counting the number of permutations (orders) of objects. If *m* and *n* are the same, as they were in the problem at the beginning of this article, we have

*n***P***n*  = *n*!(*n* – *n*)! = *n*!0!.

To make sense of this we define 0! = 1. Now *n***P***n* = *n*! as we would expect from our solution to the first problem.

**EXERCISE**

**SOLUTION**



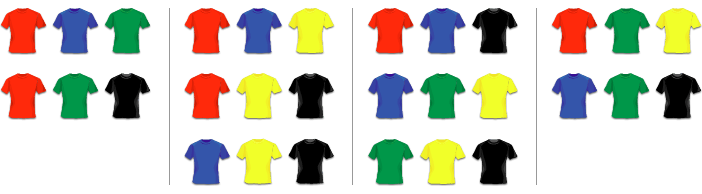
Unfortunately you can’t remember the code for your four-digit lock. You only know that you didn’t use any digit more than once. How many different ways do you have to try? What do you conclude about the safety of those locks?

**Combinations**

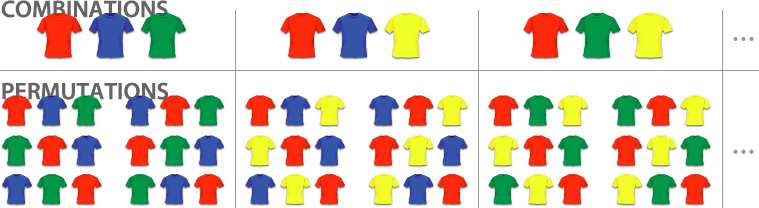
Permutations are used when you select objects and care about their order – like the order of children on chairs. However in some problems you don’t care about the order and just want to know how many ways there are to select a certain number of objects from a bigger set.

*In a shop there are five different T-shirts you like, coloured red, blue, green, yellow and black. Unfortunately you only have enough money to buy three of them. How many ways are there to select three T-shirts from the five you like?*

Here we don’t care about the order (it doesn’t matter if we buy black first and then red or red first and then black), only about the number of **combinations** of T-shirts. The possibilities are



so there are 10 in total. If we had calculated 5**P**3 = 60, we would have double-counted some possibilities, as the following table shows:



With permutations, we count every combination of three T-shirts 6 times, because there are 3! = 6 ways to order the three T-shirts. To get the number of combinations from the number of permutations we simply need to divide by 6. We write

5**C**3  =  5**P**33!  =  606  =  10.

Here the **C** stands for “**c**ombinations”. In general, if we want to choose *r* objects from a total of *n* there are

*n***C***r*  =  *n***P***rr*!  =  *n*!*r*! (*n* – *r*)!

different combinations. Instead of *n***C***r* mathematicians often write *n***C***r* = (*nr*), like a fraction in brackets but without the line in between. (To simplify typesetting we will continue using the first notation inline.)

**EXERCISES**

**SOLUTIONS**



(a) There are 10 children in your class but you can invite only 5 to your birthday party. How many different combinations of friends could you invite? Explain whether to use combinations or permutations.

(b) At a party there are 75 people. Everybody shakes everybody’s hand once. How often are hands shaken in total? *Hint: How many people are involved in shaking hands?*

**Combinatorics and Pascal’s Triangle**

Let’s calculate some values of *n***C***r*. We start with 0**C**0. Then we find 1**C**0 and 1**C**1. Next, 2**C**0, 2**C**1 and 2**C**2. Then 3**C**0, 3**C**1, 3**C**2 and 3**C**3. We can write down all these results in a table:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | 0**C**0 = 1 | |  |  |  |  |  |
|  | | | | 1**C**0 = 1 | | 1**C**1 = 1 | |  |  | | |
|  | | | 2**C**0 = 1 | | 2**C**1 = 2 | | 2**C**2 = 1 | |  |  | |
|  | | 3**C**0 = 1 | | 3**C**1 = 3 | | 3**C**2 = 3 | | 3**C**3 = 1 | |  |  |
|  | 4**C**0 = 1 | | 4**C**1 = 4 | | 4**C**2 = 6 | | 4**C**3 = 4 | | 4**C**4 = 1 | |  |
| 5**C**0 = 1 | | 5**C**1 = 5 | | 5**C**2 = 10 | | 5**C**3 = 10 | | 5**C**4 = 5 | | 5**C**5 = 1 | |

This is exactly Pascal’s triangle which we explored in the article on [sequences](https://mathigon.org/world/Sequences). It can be created more easily by observing that any cell is the sum of the two cells above. Hidden in Pascal’s triangle there are countless patterns and number sequences.

pascal

Now we also know that the *r*th number in the *n*th row is also given by *n***C***r* (but we always have to start counting at 0, so the first row or column is actually the zeroth row). If we apply what we know about creating Pascal’s triangle to our combinations, we get

(*nr*) + (*nr* + 1) = (*n* + 1*r* + 1) .

This is known as **Pascal’s Identity**. You can derive it using the definition of *n***C***r* in terms of factorials, or you can think about it the following way:

*We want to choose r + 1 objects from a set of n + 1 objects. This is exactly the same as marking one object of the n + 1, to be called X, and either choosing X plus r others (from the remaining n), or not choosing X and r + 1 others (from the remaining n).*

Many problems in combinatorics have a simple solution if you think about it the correct way, and a very complicated solution if you just try to use algebra…

**STARS AND BARS**

**SOLUTION**

**EXAMPLE**

A greengrocer on a market stocks a large number of *n* different kinds of fruit. In how many ways can we make up a bag of *r* fruits? Note that *r* can be smaller, equal or bigger than *n*.

**Combinatorics and Probability**

Combinatorics has many applications in [probability theory](https://mathigon.org/course/probability). You often want to find the probability of one particular event and you can use the equation

*P*(*X*)  =  probability that *X* happens  =  number of outcomes where *X* happenstotal number of possible outcomes

You can use combinatorics to calculate the “total number of possible outcomes”. Here is an example:

*Four children, called A, B, C and D, sit randomly on four chairs. What is the probability that A sits on the first chair?*

We have already shown that in total there are 24 ways to sit on four chairs. If you look back at our solution, you will also find that A sits on the first chair in six of the cases. Therefore

*P*(A sits on the first chair)  =  number of outcomes where A sits on the first chairtotal number of possible outcomes  =  624  =  14.

This answer was expected, since each of the four children is equally likely to sit on the first chair. But other cases are not quite as straightforward…

**EXERCISES**

**SOLUTIONS**