Sequence – notes and examples

This is a web page about sequences.

Please report bugs and send comments and suggestions to w.w.milner@gmail.com

Because of the small screen size, it would never be much use on a mobile phone, so it is intended for use on a laptop or desktop. It should work with most modern browsers.

Table of Contents

Quick start

Click 'Set and run'. See the sequence t_n=n+1, on a graph and in the table.

What is a sequence?

Here are three examples:

{3,5,7,9}

{9,1,8,2,9,1,8}

{2,4,8,16,32..}

The first sequence is the odd numbers 3 to 9 inclusive. Sequences are often written with this { curly bracket } notation.

The second example {9,1,8,2,9,1,8} shows that sequences can involve repeats (2 9s), do not have to be in increasing order, and do not have to follow *any* pattern.

The third example {2,4,8,16,32..} 'goes on forever'. We might say it is an *infinite sequence*, while the first two were finite sequences.

So as sequence is like a set, except that it is ordered and can have repeats.

Being ordered means the order matters, so that {1,2,3} is a different sequence from {3,2,1}. It does not mean it must be in increasing order.

The elements of a sequence can be drawn from any set. {3,2,1} for example has elements which are *natural numbers*, elements of the set ℕ. We could also use real numbers, such as {3.5, -7.2, 11.4}. Or complex numbers. Or not numbers at all – such as groups, vectors, matrices, functions and so on.

Indexes

Because the order matters, we often talk about the first element, the second, the third and so on. We are matching up each element with a member of ℕ. For example

Sequence	2	4	8	16	32	64	128	..
Index	1	2	3	4	5	6	7	..

So the third element of {2,4,8,16,32..} is 8.

We usually call the index ‘n’.

Then for some sequences we can find some pattern, and write the n^{th} term as a formula in n. In this case we would say $t(n) = 2^n$. We are writing t(n) as the nth term, and 2^n lets us work it out.

We then have a function from ℕ to the sequence. So 1 → 2, 2 → 4, 3 → 8, 4 → 16 and so on. It is a function, because one n cannot map to 2 different sequence terms.

But some sequences do not have a formula. {3.5, -7.2, 11.4} does not have a formula.

The terms of an infinite sequence might become very close to some value, L, as n becomes very large. We say the sequence *converges* to L, and that L is the limit of the sequence. The actual definition of this idea is called the *epsilon-delta definition*.

Some infinite sequences do not converge to a limit. An example is $(-1)^n$.

Series

If we add up the first n terms of a sequence (called the *partial sum*), we have a *series,* which is another sequence in terms of n.

For example, the sequence $t_n=n+1$

n	t_n	Sum to n
1	2	2
2	3	5
3	4	9
4	5	14
5	6	20
6	7	27
7	8	35

So now we have another sequence: 2,5,9,14,20,27.. The n^{th} term is n/2(3+n). This is our series, S_n = n/2(3+n)

Some sequences (like 1/n) have terms that converge to 0. But the corresponding infinite series, the sum of 1/n to infinity, does not converge. $1/n^2$ does.

Defining a sequence

An expression for the n^{th} term can be entered, using the syntax described below.

The initial terms of the sequence can be entered, as values separated by commas. So for example if we enter 4,5,6 here, then we will have

t_1=4 t_2=5 t_3=6

Initial terms can include parameters, such as a+3, a+4

Recurrent definitions

t[n-1] is the previous term, t[n-2] is the one before that, and so on. Check the [square brackets].

So the usual Fibonacci sequence, for example, is t[n-1]+t[n-2], with initial terms 1,1

Errors

Invalid expression include unmatched brackets, like 2*(n+1, or repeated operators like a+ +3 and so on.

If an error is detected, it will show 'Problem with n++' or whatever at the top, and things will stop working.

Just re-enter the corrected expression and carry on.

Expression syntax

This is like normal maths, using (round brackets), except * is needed for multiplication. Examples are

2n+3 is written 2*n+3

sin(4n) is written sin(4*n)

$\frac{2n+1}{n-1}$ is written (2*n+1)/(n-1)

and so on.

Single lower case letters are parameters mostly set to 1 (but can be changed). e is 2.71.. and pi is π.

x*x and pow(x,2)

pow(a,b) is a^b, so x*x and pow(x,2) should work out the same.

But pow(a,b) works by finding the log of a, multiplying by b, then finding exp of the result. It is therefore slow and not very accurate.

For squares and cubes, x*x and x*x*x are faster and more accurate.

Built in functions

Deja knows about several functions already.

Common functions

sin, abs (absolute value), acos (inverse cosine), atan, ceil (ceiling function), exp (exponential), floor, log (natural), pow (power, so pow(x,-2) = x^{-2}), sqrt, tan, cos, acosh, asin, atanh, cbrt (cube root), cosh, sinh, tanh.

The trig functions use radians.

Less common functions

gamma(x) : gamma function

laguerre(n,x) : the nth Laguerre polynomial

fact(n) : n!

hermite(n,x) : the nth Hermite polynomial

ddx(n, exp, x) : the nth differential coefficient, of a function exp, written as a string in quotes. So for example ddx(2,"x*x*x",x) = $\frac{d^2}{dx^2}x^3$

integral(exp,x) : the definite integral of a function exp, written as a string, with the lower limit being xMin, the left-hand edge of the graph.

leg(n,x) : The nth Legendre polynomial

poch(n,x) : The Pochhammer function, the rising factorial. So poch(3,x) = x(n+1)(n+2)

sq(x) : A square wave

tri(x) : A triangular wave.

Parameters

Parameters a to z can be referred to in expressions. These are initially all 1, but this can be changed by ‘Set parameter’

Examples

Constant sequence

t[n]=7

Pretty dull – but it is a sequence

Arithmetic progression

t(n) =a+(n-1)*d

a=first term, d='common difference'

Recurrent definition : t[1]=a, t[n]=t[n-1]+d

Geometric progression

t[n]=a*r**(n-1)

Change a and r from 1 for a more interesting sequence.

Recurrent definition: t[1]=1, t[n]=t[n-1]*r

Harmonic progression

1/arithmetic progression:

t[n]=1/(a+(n-1)*d)

Harmonic series

t[n]=1/n

sin(nπ/2)

t[n]=sin(nπ/2)

If n=1,5,9.. t is 1. For 3,7,11.. -1. For 2,4,6.. 0.

sin(n)

t[n]=sin(n)

Chaos

t[n]=4*t[n-1]*(1-t[n-1])

This is the 'logistic equation', and is the classic example of chaotic behaviour. The sequence depends on the initial value, between 0 and 1.

Leibnitz π

t[n] = 4*(-1)**(n-1)/(2*n-1)

The sum converges to pi (slowly)

1 – 1/3 + 1/5 – 1/7..

is pi/4.

e

t[n]=1/fact(n)

because e = $\sum_{0}^{\infty} 1/n!$