–Bignum aritmetika

az aktuálisan használt stuktúrában pontosan számol

eldöntik melyik struktúrába tartozik a szám

abban számolnak

Maple típusok

egész, racionális, komplex, vagy efölötti polinom

típusok mindig algebrai struktúrák, halmazok, ill. műveletek eredménystruktúrái

Egészek

GMP könyvtár

helyiértékes reprezentácó

dinamikus adatvektor

kis egészek

számrendszer alapszáma 10000

Függvények

lnko kibővített lnko prímtesztelés

maradékos osztás

prímfaktorizáció

Racionális számok

egészeket összenyilaz

automatikus egyszerűsítés

számláló és nevező numer(), demon() függvényekkel kezelhető

köztes tárrobbanás

lebegőpontos számok

mantissza karakterisztika

Float(10,-7)

Dupla pontosságú lebegőpontos

RR gyűrű elemei

Algebrai számok

nem alapvető típus

RootOf procedúrával

Komplex számok

a+b\*I vagy konstruktorral Complex(a,b)

Re(), Im(), conjugate(), abs(),argument()

csgn()

Maple értékadás

name:=expr;

(name1,..namen):=(expr1,…exprn);

tesztelés assigned(name)

hozzárendelés visszavonás name:=`name` unassign(`name`1`…)

assing(a,b) assign(A=B), assign(L)

a,b paraméterek a=b egyszerű változókra

L egyenletek listája

Teljes kiértékelés:

általános kiértékelési stratégia

utolsónév kiértékelés linalg csomag

Relációs műveletek <>

Logikai and or xor implies not

Log. nevek true false FAIL =>háromértékű logika

Ciklus

for from by to do

from by to tetsz. sorrend

for in

while do end do

Értékadás

subs vagy eval

Függvén def.

fn:=paraméterek->kif.

Eljárás

proc end proc

param. típusegyeztetés ::-vel

ha nincs return utoljára számított megy vissza

rekurzió option remember

Függvény kompozíció f@g f@@n (önmagával) f@@(-1) inverz

Funkcionális progr.

map, map2, zip, curry, uncurry

Nevek

\_ kezdődik

tetszőleges számú betű száámjegy \_ követi

case sensitive

Spec. karakterek

alpha,..,Alpha,..

protect

nevek levédése

macro(név=kif)

lnko

gcd(p1,p2) bővített gcdex

lkkt lcm

faktorizáció

factor(p) Z Q felett

factor(p,K) K testbővítés

maradékosztályokban Factor(p) mod m

Racionális törtfüggvény numer,denom

nincs automatikus egyszerűsítés

egyszerűsítés: normal

Határérték

limit(f(x),x=a, direction)

Polinomok

minden kifejezés egy DAG

FLINT, NTL

Racionális kifejezésként kezel

trigonometrikus függvények polinomjai

típusa ratpoly(integer, sin(x))

subs kifejezés helyettesítése

Mintaillesztés

applyrule

**sssssssssssssssssssssssssssssssssssssssssssssss**

Maple

Különböző egyenletek megoldása

fsolve: egyenletek közelítő megoldására paraméterei

Egyenletrendszer

ismeretlen, amire meg akarjuk oldani

complex ha komplex megoldásokat keresünk

Pl

|  |
| --- |
| polynomials:≔{3*x*4*y*2=17,*x*3*y*−5*xy*2−2*y*=1}: |
|  |

|  |  |
| --- | --- |
| **>** | fsolve(polynomials) |

isolve

egész számok körében oldja meg az egyenletrendszert

NULL-t ad vissza ha Maple nem talál egész megoldást

Z1 itt bármilyen egész lehet 2. paraméterben Z1 helyett megadhatunk bármit

|  |
| --- |
| isolve(3*x*−4*y*=7) |

|  |
| --- |
| {x=5+4\_Z1,y=2+3\_Z1} |
|  |
|  |
|  |
|  |

rsolve

paraméterek rekurzív függvény meglehet adni f-nek kezdőértékeket

második paraméter ami alapján számolja az értéket

rsolve({*f*(*n*)=−3*f*(*n*−1)−2*f*(*n*−2),*f*(1..2)=1},{*f*})

dsolve

differenciál egyenletek megoldására

első paraméter egyenlet második feltételek a függvényre

|  |  |
| --- | --- |
|  | ode≔d2dx2y(x)=2y(x)+1 |
|  | ics≔*y*(0)=1,D(*y*)(0)=0 |

dsolve({ode,ics})

solve

egyenletek, egyenletrendszerek megoldása

paraméter egyenletrendszer, ismeretlen amire meg akarjuk oldani

Gráfok

Irányítatlan G

Irányított H

Ábr. DrawGraph(G)

SpecialGraphs[PetersenGraph]()

ChromaticPolynomial(P,lambda)

NonIsomorphicGraphs

paraméterek: csúcsok száma, élek száma

hány ilyen nem izomorf gráf

IsIsomorphic(g,h)

izomorfak-e a gráfok

MinimalSpannigTree(G)

KruskalAlgorithm(G,animate)

PrimsAlgorithm(G,animate)

BellmanFordAlgorithm(gráf,gráf csúcsai)

DijkstrasAlgorithm

ShortestPath

TopologicSort(gráf)

TravelingSalesman(G)

|  |
| --- |
|  |
|  |

|  |
| --- |
|  |

Listák

hossza numelems(L)

lista tartalma op(L)

elem hozzáfűzése a lista végére L:=[op(L),elem]

adott indexű elem törlése L:=subsop(index=NULL,L)

elem tagja-e a listának member(elem,L)

Halmazok

s1 union s2

s1 minus s2

s1 intersect s2

s1 subset s2

{1,2,3}

1 in {1,2,3} eleme-e

Vectorok

Vector(1..3,5) [5,5,5] oszlopvektor

Vector[row]([1,2,3])

ScalarMultiply(A,s) vektor szorzása skalárral

Mátrixok

egységmátrix Matrix(3,shape=identity)

Matrix(2) 2x2-es null mátrix Matrix(2,3) has.

**A + B**

**A . B**

**x \* A**

**A \* x**

**A ^ n**

**A ^ +**

**&\* mátrix szorzás**

**Transpose(A)**

m2≔Matrix([[1,2,3],[4,5,6]])

|  |  |
| --- | --- |
|  | *A*≔Array([1,2,3]) |

ArrayDims(*A*)

upperbound(*A*)

sorok száma ArrayNumDims(*B*)

ArrayNumElems(*B*)

rtable: Mátrix, vektor, array építésére

Határérték

limit(f(x),a,direction): f(x) fv. a helyen vett bal vagy jobb oldali határértéke

visszaadott érték lehet intervallum is

direction left right complex,real

Deriválás

diff(expr,x,x...,x) vagy diff(expr,x$n)

D [i] (függvény)

Határozatlan integrál

int(expr,var)

Határozott integrál

int(expr, range)

Numerikus integrálás

evalf(Int(expr, range[list][, method, epsilon, digits])